

BOGDAN TOMOYUKI NASSU

**DESCOBERTA DA TOPOLOGIA DE REDES
DINÂMICAS E DESCENTRALIZADAS COM
AGENTES MÓVEIS INTELIGENTES**

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre. Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, Universidade Federal do Paraná.

Orientador: Prof. Dr. Elias P. Duarte Jr.

CURITIBA

2005

Agradecimentos

Agradeço a todos que me ajudaram nesses meses nos quais estive trabalhando para escrever essa dissertação. Agradeço pelas idéias, mesmo aquelas que aparentemente nada tinham a ver com o trabalho, e pelo apoio. Agradeço aos meus amigos, pessoas gordas e hilárias, que sempre me alimentam com conversas que expandem os meus horizontes. Agradeço à minha mãe, que sempre esteve me ajudando com tudo aquilo que eu não podia fazer, por estar ocupado demais trabalhando ou cansado demais depois de ter trabalhado. Agradeço aos meus professores da graduação, especialmente o Urban e o Michel, por terem me ensinado que o todo é maior que a soma das partes e que agentes inteligentes podem ser bem interessantes. Agradeço à professora Cristina pelas boas idéias que me ajudaram a concluir este trabalho em bem menos tempo que o previsto. Agradeço ao professor Elias, apesar disso ser meio redundante, já que o seu nome está ali na capa creditado como “orientador” — agradecer ao orientador, neste caso, seria como agradecer a mim mesmo. E deixo um agradecimento especial à Fernanda Li Minku, minha querida amiga, que sempre me ofereceu um ombro amigo nas horas de dúvida e desânimo, além de ter me apresentado ao “teste-t” que valorizou — e muito — o meu trabalho.

Resumo

Neste trabalho, é proposta uma estratégia para a descoberta distribuída da topologia de redes dinâmicas e descentralizadas, tais como redes *peer-to-peer* descentralizadas não estruturadas e redes móveis sem fio ad hoc. Na estratégia proposta, agentes móveis — entidades computacionais autônomas, capazes de se deslocar entre os nodos de uma rede — são usados para disseminar informações a respeito da topologia da rede. Esta estratégia se baseia em uma metáfora inspirada em colônias de formigas, na qual os agentes são simples e se comunicam através da estigmergia — forma de comunicação usada por formigas reais. Esta metáfora se fundamenta nos princípios da inteligência coletiva, ou seja, a inteligência e complexidade do sistema não estão nos agentes individuais, mas sim no produto das suas interações. Diversos aspectos da estratégia proposta são apresentados e discutidos, tais como a confiabilidade das informações frente o dinamismo da rede, os métodos usados para determinar os itinerários percorridos pelos agentes, e a definição do número de agentes que deve usado em redes cujo tamanho pode variar com o passar do tempo. Resultados experimentais mostram que a estratégia produz descrições da topologia bastante similares à topologia real. Eles também demonstram que o método baseado em feromônios para a seleção dos destinos dos agentes melhora o desempenho da estratégia em comparação com uma seleção aleatória; e que a população de agentes pode ser ajustada dinamicamente às mudanças de tamanho da rede.

Abstract

This work introduces a strategy for topology discovery in decentralized and dynamic networks, such as decentralized and unstructured peer-to-peer networks and mobile ad hoc networks. In the proposed strategy, mobile agents — autonomous computational entities that are able to move between the nodes in a network — are used to disseminate information about the network topology. This strategy is based upon a metaphor inspired by ant colonies, in which the agents are simple and communicate through stigmergy — a communication scheme used by real-world ants. This metaphor follows the ideas of swarm intelligence, meaning the system’s intelligence and complexity are not in the agents themselves, but in the product of their interactions. Several aspects of the proposed strategy are presented and discussed, such as the reliability of the topology information given the dynamic nature of the network, the methods used by the agents to determine their itineraries, and the number of agents that are used in networks that have an arbitrary size. Experimental results show that the strategy produces topology descriptions that are very close to the real topology. They also show that the pheromone-based method for agent destination selection attains better results than a random selection, and that the number of agents can be automatically adjusted as the size of the network changes.

CONTEÚDO

Resumo	ii
Abstract	iii
1 Introdução	1
2 Redes Dinâmicas e Descentralizadas	4
2.1 Redes <i>Peer-to-Peer</i> Descentralizadas Não Estruturadas	5
2.2 Redes Móveis Sem Fio Ad Hoc	9
3 Agentes Móveis Inteligentes	15
3.1 Agentes Inteligentes e Sistemas Multiagentes	16
3.1.1 Sistemas Multiagentes	19
3.2 Agentes Móveis	21
3.2.1 Mobilidade	23
3.2.2 Comunicação entre Agentes Móveis	26
3.2.3 Uma Arquitetura para um Sistema de Agentes Móveis	32
3.3 Inteligência Coletiva e Sistemas Baseados em Colônias de Formigas	34
3.3.1 Inteligência Coletiva	35
3.3.2 Sistemas Baseados em Colônias de Formigas	36
4 Trabalhos Relacionados	40
4.1 Agentes Móveis para o Gerenciamento de Redes	40
4.2 Agentes Móveis para o Roteamento	42
4.3 Agentes Móveis para a Localização de Recursos em Redes P2P	44

4.4	A Estratégia de RoyChoudhury, Paul e Bandyopadhyay para a Descoberta da Topologia	46
5	Uma Estratégia para a Descoberta da Topologia de Redes Dinâmicas e Descentralizadas com Agentes Móveis Inteligentes	49
5.1	Modelo da Rede	49
5.2	Funcionamento Básico	51
5.3	Desatualização das Informações	52
5.4	Idade e Envelhecimento das Informações	54
5.5	Migração	55
5.6	População de Agentes de Descoberta	57
5.7	Tempo Antes de Migrar	59
5.8	Alcance da Descrição	61
6	Avaliação Experimental	64
6.1	ANTD (<i>Agents for Network Topology Discovery</i>)	65
6.1.1	Topologia Inicial	66
6.1.2	Dinamismo	68
6.1.3	Configuração da Estratégia para a Descoberta da Topologia	70
6.1.4	Métricas	72
6.2	Experimentos e Resultados	75
6.2.1	Qualidade das Descrições da Topologia	78
6.2.2	Dois Cenários Mais Complexos	83
6.2.3	Idade e Envelhecimento das Informações	85
6.2.4	Migração	89
6.2.5	População de Agentes de Descoberta	94
6.2.6	Tempo Antes de Migrar	98
6.2.7	Conclusões dos Experimentos	100
7	Conclusão	102
	Referências Bibliográficas	104

Capítulo 1

Introdução

A popularização de novas tecnologias tem feito surgir um grande interesse por diferentes arquiteturas para redes de computadores. O aumento da capacidade das conexões à Internet faz com que o número de usuários de redes *Peer-to-Peer* (P2P) [79, 77, 75] para o compartilhamento de arquivos cresça a cada dia. Além disso, equipamentos portáteis com interfaces para a comunicação sem fio tornam possível a criação de redes móveis ad hoc [19], nas quais não existe qualquer tipo de infra-estrutura previamente preparada. Entretanto, esses tipos de rede possuem diversas características, como as mudanças constantes — dinamismo — e a ausência de um controle central — descentralização — que criam uma série de desafios que devem ser superados.

Em redes P2P descentralizadas não estruturadas e redes móveis sem fio ad hoc, aplicações que normalmente se baseiam em abordagens centralizadas devem procurar estratégias distribuídas e descentralizadas. Entre essas aplicações, pode-se destacar a descoberta da topologia da rede. O conhecimento sobre a topologia da rede é útil, ou mesmo essencial, para várias outras aplicações, como sistemas para gerenciamento [50] e roteamento [20]. Neste trabalho, é proposta uma estratégia para a descoberta da topologia de redes dinâmicas e descentralizadas baseada em agentes móveis inteligentes.

Agentes móveis são entidades computacionais autônomas, localizadas em uma rede, e capazes de se deslocar entre os seus nodos [45]. Em ambientes complexos, como as redes consideradas neste trabalho, agentes podem precisar agir com uma certa flexibili-

dade, ou inteligência [97], cooperando com outros agentes para satisfazer seus objetivos. Neste trabalho, os agentes são modelados de acordo com uma metáfora inspirada em colônias de formigas [47, 101], na qual os agentes são simples e se comunicam através da estigmergia [42, 103, 48] — forma de comunicação indireta usada por formigas reais. Os sistemas baseados em colônias de formigas seguem os princípios da *swarm intelligence* [9, 61], ou inteligência coletiva, propriedade de sistemas compostos por agentes simples que exibem coletivamente um comportamento complexo, com a inteligência do sistema não se localizando nos agentes individuais, mas no produto das suas interações.

Na estratégia proposta por este trabalho, agentes se deslocam entre os nodos da rede, disseminando informações que descrevem a sua topologia. O modelo de rede considerado engloba diversas características de redes como as redes ad hoc e as redes P2P descentralizadas não estruturadas — mas não se restringe apenas a estes casos. A estratégia proposta tem vários aspectos que devem ser considerados, como a confiabilidade das informações em uma rede que está em constante transformação, as técnicas que os agentes usam definir os seus itinerários, e o número de agentes usado em uma rede que possui um tamanho que é, a princípio, indeterminado e variável.

Este trabalho se divide da seguinte forma. No capítulo 2 são descritas as redes consideradas: redes P2P descentralizadas não estruturadas (seção 2.1) e redes móveis sem fio ad hoc (seção 2.2). No capítulo 3, são abordados os sistemas compostos por agentes móveis inteligentes. A seção 3.1 descreve os agentes inteligentes e os sistemas multiagentes. Na seção 3.2, são discutidas diversas questões referentes ao uso de agentes móveis. A seção 3.3 apresenta a inteligência coletiva e os sistemas baseados em colônias de formigas. No capítulo 4, são descritos vários trabalhos relacionados: aplicações baseadas em agentes móveis para o gerenciamento de redes (seção 4.1), roteamento (seção 4.2) e localização de recursos em redes P2P (seção 4.3); assim como uma estratégia para a descoberta da topologia de redes ad hoc (seção 4.4).

No capítulo 5, a estratégia proposta por este trabalho para a descoberta da topologia é detalhada. Um modelo de rede é definido na seção 5.1, e o funcionamento básico da estratégia é descrito na seção 5.2. As seções 5.3 a 5.8 tratam de diversos aspectos da

estratégia proposta. O capítulo 6 traz a avaliação da estratégia proposta. São descritos os ANTD, simulador desenvolvido para a avaliação (seção 6.1), e os experimentos realizados, junto com os resultados obtidos (seção 6.2). Por fim, o capítulo 7 traz a conclusão deste trabalho.

Capítulo 2

Redes Dinâmicas e Descentralizadas

Neste trabalho, define-se uma rede como um sistema composto por diversas unidades de processamento, chamadas de nodos, conectadas entre si por enlaces de comunicação. Dois nodos são ditos vizinhos um do outro quando há um enlace entre eles. A topologia de uma rede é a forma como esses enlaces se configuram física ou logicamente [34]. Uma topologia física se refere às conexões reais existentes entre os nodos, feitas através de cabos, antenas ou outros meios físicos. Uma topologia lógica se refere às conexões virtuais, que são obtidas através do uso de protocolos de comunicação. A topologia de uma rede pode ser mapeada para um grafo cujos vértices e arestas representam, respectivamente, os nodos e os enlaces.

O conhecimento da topologia de uma rede pode ser útil, ou mesmo essencial, em aplicações tais como o diagnóstico do estado das unidades [58], a localização de falhas em nodos [60] ou sistemas de gerenciamento de rede [50]. O tamanho e complexidade das redes tornam necessário o uso de técnicas para a descoberta automática de suas topologias. Esta tarefa é normalmente realizada por um nodo central, que se utiliza de ferramentas baseadas em protocolos como o SNMP (*Simple Network Management Protocol*) ou protocolos proprietários [6]. Em alguns casos, é importante que todos os nodos da rede conheçam a sua topologia. Nesses casos, o nodo central pode compartilhar a sua descrição com os demais. Existem, porém, redes nas quais não é possível ou desejável determinar um nodo central. Em tais redes, podem-se adotar soluções descentralizadas,

nas quais vários nodos descobrem parte da topologia e trocam informações entre si.

Este trabalho trata de redes descentralizadas cuja topologia física ou lógica é arbitrária e dinâmica, ou seja, redes que não possuem um nodo central, nas quais não há necessariamente um canal de comunicação entre duas unidades quaisquer, e cuja topologia pode variar com o passar do tempo. Entre estas redes estão as redes *peer-to-peer* (P2P) descentralizadas não estruturadas e as redes móveis sem fio ad hoc. Neste trabalho, o problema da descoberta da topologia de uma rede por seus nodos tem aplicação potencial nestes casos.

Este capítulo descreve as redes P2P e as redes ad hoc. A seção 2.1 descreve as redes P2P, enquanto a seção 2.2 descreve as redes ad hoc.

2.1 Redes *Peer-to-Peer* Descentralizadas Não Estruturadas

Atualmente, a maior parte das aplicações na Internet é baseada no modelo cliente-servidor [94]. Neste modelo, os recursos — informações e serviços — são centralizados em servidores, que os disponibilizam em resposta às requisições de vários clientes. Os clientes assumem um papel passivo, consumindo recursos, mas não os disponibilizando. O aumento do número de clientes torna necessário um investimento cada vez maior em servidores mais robustos, enquanto muitas vezes o potencial computacional dos clientes não é aproveitado.

As redes *peer-to-peer* (P2P) apresentam uma maneira de disponibilizar recursos diferente do modelo cliente-servidor [81, 79]. Em uma rede P2P, todos os nodos podem disponibilizar e consumir recursos, possuindo simultaneamente características de cliente e servidor. Para realizar o compartilhamento, os nodos se conectam diretamente uns aos outros. Os vizinhos de um nodo são os seus *peers* — entidades com características similares às do próprio nodo.

As redes P2P consideradas neste trabalho são redes *overlay* construídas sobre a Internet. Uma rede *overlay* é uma rede cuja topologia lógica é construída sobre uma outra rede já existente [81]. A Internet em si é uma rede deste tipo, pois é uma rede virtual construída sobre diversas redes que usam tecnologias variadas.

Exemplos de aplicações que se baseiam em redes P2P incluem aplicações para troca

de mensagens instantâneas, como o ICQ [51] e o MSN Messenger [22], e aplicações para compartilhamento de arquivos, como o Gnutella [77], o KaZaA [75] e o Napster [74]. Existem também *frameworks* que oferecem suporte direto para a implementação de redes P2P, tais como o JXTA [72] e o .NET [36].

Não existe consenso quanto à definição de rede P2P, nem tampouco quanto à sua classificação, com diversas propostas podendo ser encontradas na literatura. Uma possível classificação das redes P2P [67] as divide em três categorias, descritas abaixo: centralizadas, descentralizadas estruturadas, e descentralizadas não estruturadas.

Uma rede P2P centralizada possui um ou mais servidores centrais, responsáveis pela coordenação dos nodos ou pela manutenção de índices de busca. No primeiro caso, os servidores atribuem tarefas e responsabilidades aos nodos, que passam a agir com autonomia. No segundo caso, ilustrado na figura 2.1, os servidores possuem índices que são mantidos atualizados, com os recursos disponibilizados pelos nodos sendo publicados para que outros nodos possam encontrá-los. Uma vez que um nodo encontra um recurso, ele se conecta a nodos que o tenham para fazer a requisição. Esse modelo é adotado, por exemplo, pelo Napster [74].

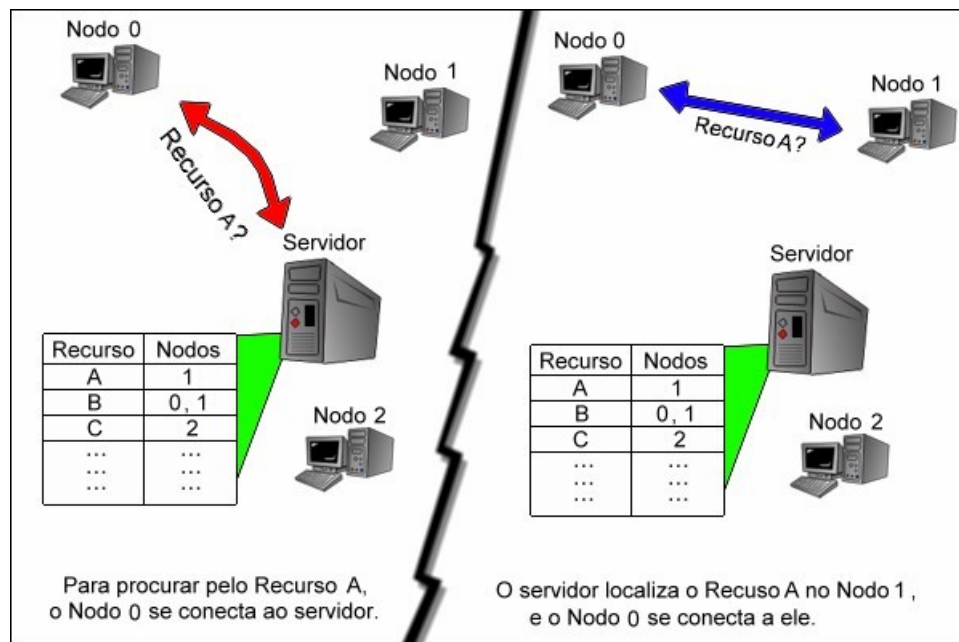


Figura 2.1: Uma rede P2P com índice centralizado.

Redes P2P centralizadas podem não ser muito escaláveis, com o gargalo se localizando

nos servidores centrais. Além disso, problemas que tornem indisponíveis estes servidores podem interromper o funcionamento de toda a rede. Por fim, a rede permanece sob o controle direto das entidades que mantêm os servidores — o que pode ser tanto uma vantagem como uma desvantagem.

Uma rede P2P descentralizada estruturada não possui servidores centrais, mas tem um alto grau de estruturação, com a atribuição de responsabilidades especiais a certos nodos, a topologia sendo controlada, ou os recursos sendo posicionados de maneira estratégica na rede. Desta forma, são possíveis a obtenção de um balanceamento na distribuição dos recursos e a criação de algoritmos de busca eficientes, que tiram proveito da estruturação da rede [17]. O Freenet [38] e o KaZaA [75] são exemplos de redes P2P descentralizadas estruturadas. Uma rede deste tipo, na qual alguns “supernodos” recebem a responsabilidade de gerenciar as buscas de um grupo de nodos, é ilustrada na figura 2.2. Os nodos se conectam aos supernodos e estes se conectam entre si para realizar buscas. O compartilhamento de recursos é então feito diretamente entre os nodos. Os supernodos também podem compartilhar recursos, como qualquer outro nodo.

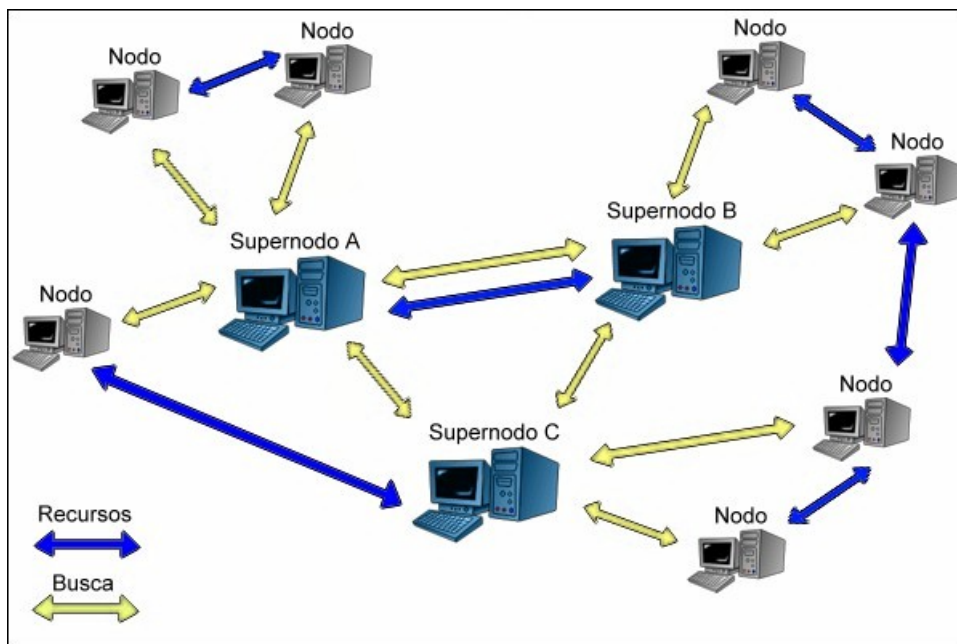


Figura 2.2: Uma rede P2P descentralizada estruturada na qual supernodos gerenciam as buscas de outros nodos.

Em situações nas quais a estruturação não é possível ou desejável, podem-se usar redes

P2P descentralizadas não estruturadas. Uma rede deste tipo não possui qualquer diretório centralizado e não mantém um controle rígido sobre a topologia ou o posicionamento dos recursos. Esse é o tipo de rede P2P considerado neste trabalho. A rede P2P descentralizada não estruturada mais conhecida é a rede Gnutella [77]. A rede formada apenas pelos supernodos em certas redes P2P descentralizadas estruturadas também pode ser considerada uma rede deste tipo. A figura 2.3 mostra um exemplo de rede P2P descentralizada não estruturada. Neste exemplo, um nodo envia mensagens de busca e compartilha recursos com os seus vizinhos.

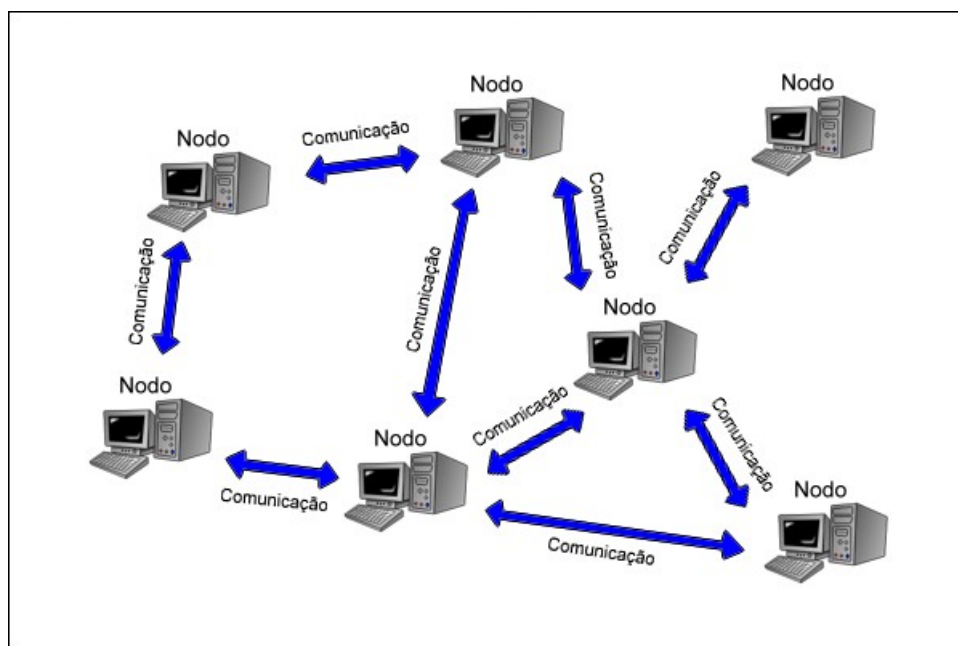


Figura 2.3: Uma rede P2P descentralizada não estruturada.

Em algumas aplicações, as redes P2P descentralizadas não estruturadas oferecem vantagens sobre o modelo cliente-servidor. Estas vantagens surgem por conta de características relacionadas à sua arquitetura distribuída e descentralizada, como a robustez, a tolerância a falhas, e a utilização da capacidade de processamento ou armazenamento de nodos ociosos. Por exemplo, uma informação pode estar distribuída ou replicada em diversos nodos, de forma que uma falha em um nodo não torna a informação necessariamente inacessível. Porém, como nem todas as aplicações se beneficiam destas características, pode-se dizer que o modelo P2P não irá substituir o modelo cliente-servidor, mas sim complementá-lo. Além disso, existem diversas questões que devem ser consideradas

quando o modelo P2P descentralizado não estruturado é usado. Abaixo são descritas algumas destas questões [79, 17, 67].

A maior parte das redes P2P possui um grande dinamismo, com conexões e desconexões ocorrendo a todo instante. Quando um nodo se conecta à rede, outros nodos devem ficar cientes da sua presença e dos recursos oferecidos. Quando um nodo falha ou se desconecta da rede, podem ocorrer particionamentos, e recursos que estavam sendo utilizados por outros nodos são perdidos. Os nodos devem possuir grande autonomia e estar preparados para essas mudanças, de forma que a rede permaneça em funcionamento mesmo quando elas ocorrerem.

Normalmente, não existem restrições quanto à capacidade dos nodos que se conectam à rede. Desta forma, pode haver uma grande diferença entre as suas capacidades de armazenamento, processamento e largura de banda. Como um recurso pode estar distribuído de maneira arbitrária, muitas vezes um nodo tem dificuldades para encontrar um *peer* que seja capaz de disponibilizar este recurso com uma determinada garantia de qualidade de serviço. Além disso, o próprio recurso pode não ter boa qualidade, o que faz com que seja necessário que a rede seja vasculhada em busca de conteúdo que tenha a qualidade desejada.

Dependendo da sua aplicação, o tamanho da rede pode variar consideravelmente. Em muitos casos, o número de nodos pode chegar aos milhares ou mesmo milhões. Por esse motivo, as soluções desenvolvidas para redes P2P devem ser escaláveis.

Outros problemas incluem diversas questões relacionadas à segurança da rede e à legalidade do conteúdo compartilhado pelos usuários, assim como possíveis inconsistências entre as diferentes versões de uma informação que esteja em vários nodos. Porém, as características relacionadas à descentralização e ao dinamismo da topologia das redes P2P descentralizadas não estruturadas são aquelas de maior importância para este trabalho.

2.2 Redes Móveis Sem Fio Ad Hoc

Nos últimos anos, computadores e outros dispositivos portáteis, tais como telefones celulares, *notebooks* e *palmtops*, têm se tornado bastante populares. Essa popularização é

acompanhada pelo desenvolvimento de tecnologias e padrões para a comunicação entre tais dispositivos, em especial para a comunicação sem fio [19], como por exemplo os padrões Bluetooth [8] e o IEEE 802.11 [76]. O aumento da capacidade de processamento destes dispositivos e da complexidade das aplicações executadas por eles tornou possível a criação de redes com os mesmos. A esse tipo de rede, que inclui equipamentos portáteis com interfaces de comunicação sem fio, dá-se o nome de rede móvel sem fio.

Atualmente, a maior parte das redes móveis sem fio se baseia na comunicação dos dispositivos móveis com uma infra-estrutura fixa, com capacidade de processamento e transmissão superiores, que provê serviços aos mesmos. Por exemplo, a comunicação entre telefones celulares é realizada através de antenas fixas ligadas a uma operadora. Da mesma forma, pode-se criar uma rede de *notebooks* que se comunicam com pontos de acesso fixos que possuem interfaces de comunicação sem fio. Esse modelo de rede é ilustrado na figura 2.4.

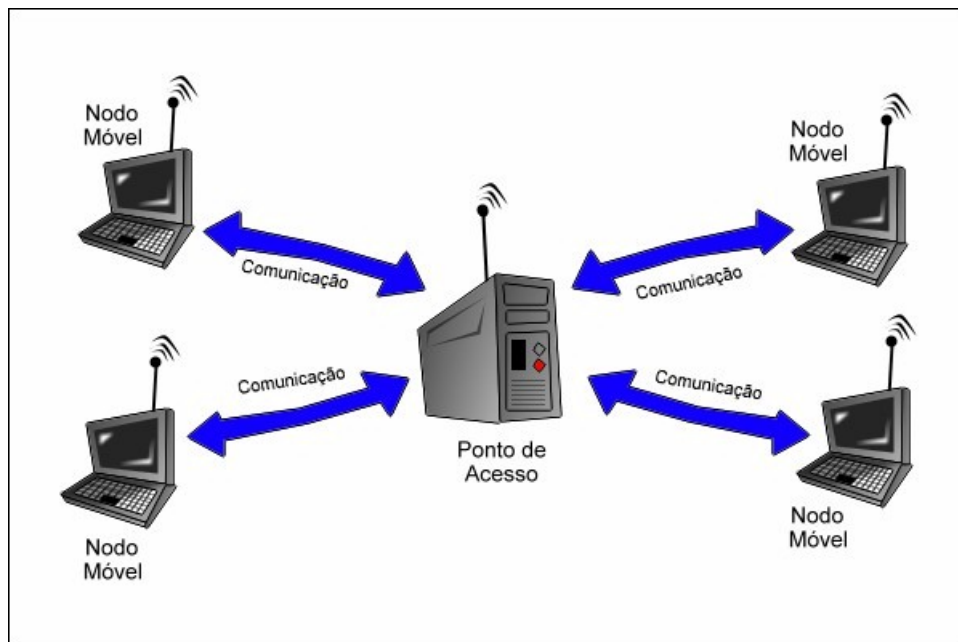


Figura 2.4: Uma rede móvel com infra-estrutura.

A preparação de uma rede móvel com infra-estrutura demanda tempo e recursos, possivelmente em quantidade significativa. Além disso, às vezes essa preparação é impossibilitada por dificuldades geográficas ou situacionais [19]. Uma alternativa capaz de resolver esses problemas é a comunicação direta entre os dispositivos móveis, sem a ne-

cessidade da preparação de uma infra-estrutura. Esse tipo de rede móvel é chamado de rede móvel sem fio ad hoc [94], ou simplesmente rede ad hoc. Uma rede ad hoc é uma rede temporária, formada dinamicamente por um conjunto de nodos localizados de forma arbitrária, com interfaces de comunicação sem fio, e que não dependem da existência de uma administração central. Esse tipo de rede é ilustrado na figura 2.5. Pode-se observar que os nodos se comunicam diretamente entre si, sem a presença de um ponto de acesso.

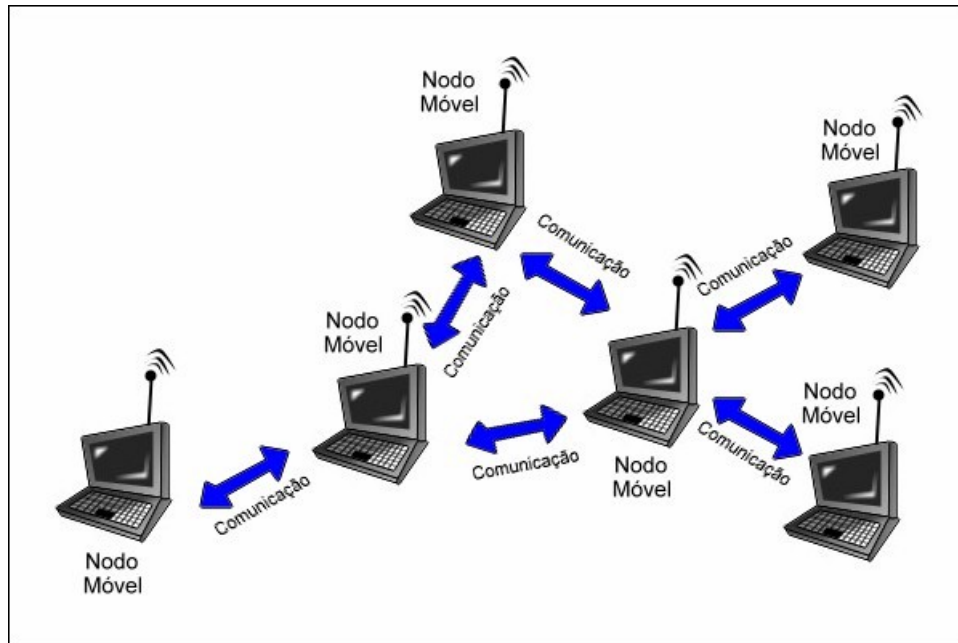


Figura 2.5: Uma rede ad hoc.

Uma rede ad hoc também pode possuir nodos fixos, ligados a outras redes como LANs ou a Internet. Neste caso, a rede é dita híbrida, e os nodos móveis podem ganhar acesso aos serviços da outra rede através dos nodos fixos. A figura 2.6 mostra uma rede móvel híbrida.

As primeiras aplicações das redes ad hoc estavam ligadas a usos militares, facilitando a comunicação e transmissão de dados entre unidades móveis em campo de batalha [37, 43]. Projetos deste tipo existem desde a década de 1970. A partir do início da década de 1990, o desenvolvimento de tecnologias comerciais para a comunicação sem fio entre dispositivos portáteis fez aumentar o interesse por usos não militares das redes ad hoc. Desde então, esta área de pesquisa vem apresentando um grande crescimento, com pesados investimentos sendo feitos no setor. Esse crescimento tende a continuar, dado o grande

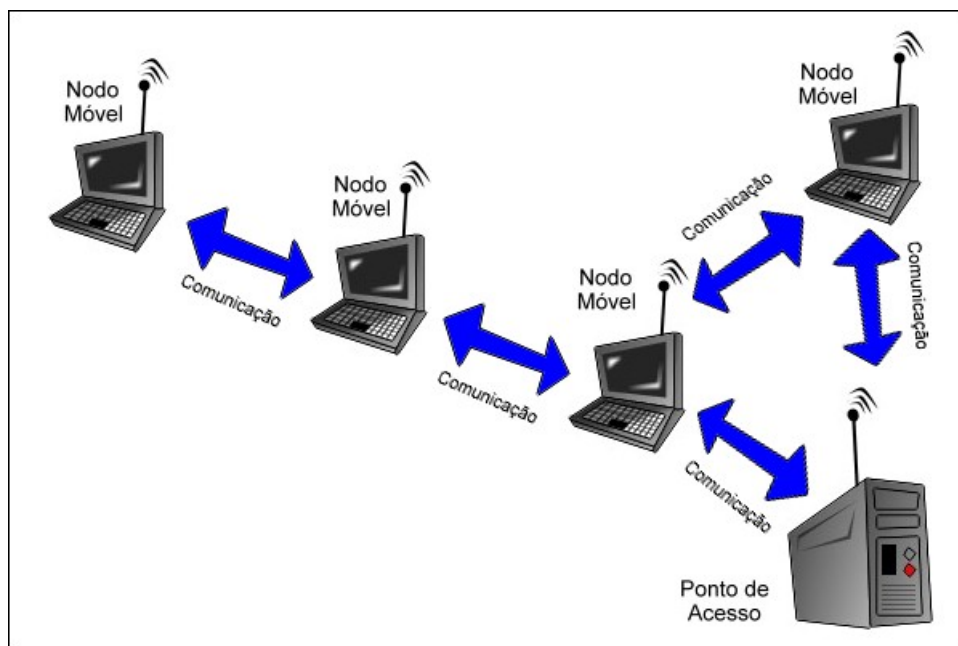


Figura 2.6: **Uma rede móvel híbrida.**

número de questões que permanecem em aberto. Recentemente, aplicações comerciais de redes ad hoc começaram a aparecer, tais como o SPANWorks [55] e o Mesh Networks [52].

A forma como são organizadas as redes ad hoc dá às mesmas uma grande flexibilidade no que diz respeito ao posicionamento dos nodos. Eles possuem liberdade de deslocamento, e podem se organizar de forma arbitrária. Essa flexibilidade é responsável por uma série de problemas, que muitas vezes impossibilitam o uso das mesmas tecnologias que são usadas em redes convencionais [19]. Além disso, os próprios nodos devem cooperar para obter a funcionalidade que normalmente é oferecida pela infra-estrutura, como roteamento, segurança e outros serviços. Entre os problemas das redes ad hoc, podem-se destacar o roteamento em múltiplos saltos, o dinamismo da topologia, a descentralização, a escalabilidade e a heterogeneidade. Estes problemas são descritos abaixo.

Em uma rede ad hoc, nodos que estejam no alcance da comunicação um do outro podem se comunicar diretamente através de tecnologias para a criação de LANs (*Local Area Networks*) sem fio [8, 76]. Se os nodos não puderem se comunicar diretamente, o seu tráfego deve ser encaminhado através de outros nodos, que atuam como roteadores. Um pacote de dados enviado por um nodo pode passar por diversos outros nodos da rede, “saltando” entre os mesmos até atingir o seu destino. A esse tipo de roteamento dá-se o

nome de roteamento em múltiplos saltos.

Como os nodos podem se mover de maneira arbitrária e imprevisível, a topologia das redes ad hoc pode variar consideravelmente com o passar do tempo. Esse dinamismo resulta na constante mudança de rotas entre os nodos, particionamentos na rede e perda de pacotes de comunicação.

A determinação de um nodo central para o gerenciamento da rede é dificultada pelo seu dinamismo. Alguns nodos podem ser escolhidos para realizar essa tarefa, mas particionamentos na rede podem isolar grupos de nodos. Isto faz com que seja importante que cada nodo da rede seja autônomo, funcionando independente de qualquer administração central. Para isto, o gerenciamento da rede deve ser distribuído entre os nodos, o que dificulta a detecção de falhas e a manutenção de um estado consistente entre os nodos [19].

O tamanho de uma rede ad hoc pode variar consideravelmente, indo desde redes de pequena escala, com até 20 nodos, até redes de larga escala, com mais de 1000 nodos [68]. Por isto, as soluções aplicadas às redes ad hoc devem ser escaláveis, o que é dificultado pela pouca disponibilidade de recursos nos nodos móveis.

Existe uma grande variedade de equipamentos que podem fazer parte de uma rede ad hoc. Pode existir uma grande diferença entre suas capacidades de processamento, memória e largura de banda. Desta forma, os nodos devem estar preparados para interagir em uma rede heterogênea. Além disso, algoritmos podem levar em conta as diferenças entre nodos para otimizar seus resultados.

A esses problemas somam-se outros, muitos dos quais são comuns às redes móveis em geral [35]. Entre estes problemas, podem-se destacar a pequena largura de banda normalmente disponível, os riscos da transmissão dos dados em um meio que é naturalmente inseguro, a pouca energia disponível nas baterias dos dispositivos móveis, assim como suas reduzidas quantidades de memória e capacidade de processamento. Existe um grande esforço de pesquisa na busca por soluções para estes e outros problemas das redes ad hoc [19], mas este trabalho se concentra nas questões relacionadas à descentralização da rede e ao dinamismo da topologia, questões estas que estão ligadas à grande mobilidade dos

nodos.

Capítulo 3

Agentes Móveis Inteligentes

O uso de técnicas de inteligência artificial tem se tornado bastante popular em aplicações que são executadas sobre ambientes complexos e imprevisíveis [25]. Em geral, essas técnicas se baseiam em heurísticas, que por vezes podem não conseguir obter resultados ótimos. Porém, elas acrescentam às aplicações um grande grau de adaptabilidade e a capacidade de obter boas soluções mesmo em situações que não haviam sido previstas no momento da sua criação. Agentes inteligentes são uma técnica derivada da inteligência artificial, possuindo estreita relação com esta área de pesquisa.

Embora não exista uma definição universalmente aceita para os agentes inteligentes, pode-se dizer que um agente inteligente é uma entidade computacional capaz de agir e tomar decisões de maneira autônoma e flexível, situada em um ambiente complexo, que normalmente tem características como o dinamismo e a imprevisibilidade [97]. Um sistema pode ser composto por diversos agentes inteligentes. Esses sistemas, nos quais diversos agentes formam uma sociedade, são chamados de sistemas multiagentes [49]. Em redes de computadores, podem-se usar agentes móveis [45], que são capazes de se mover entre os nodos.

Este capítulo descreve as características e o funcionamento dos agentes móveis inteligentes. A seção 3.1 apresenta os agentes inteligentes e os sistemas multiagentes. A seção 3.2 descreve os agentes móveis e detalha o funcionamento da mobilidade e da comunicação em sistemas baseados em tais agentes, apresentando uma arquitetura para um sistema

deste tipo. Na seção 3.3, é descrito um modelo para sistemas que procuram a inteligência coletiva, uma propriedade de sistemas compostos de agentes simples que exibem coletivamente um comportamento complexo.

3.1 Agentes Inteligentes e Sistemas Multiagentes

Aplicações em sistemas computacionais normalmente são dependentes da existência de um usuário, seja ele humano ou outra aplicação. Elas são programadas para executar ações e rotinas em resposta a comandos dados por esse usuário. Caso encontrem uma situação que não havia sido prevista no seu desenvolvimento, elas podem necessitar de uma intervenção do usuário, que determina qual deve ser a sua reação. Esse é um modelo aceitável, ou mesmo necessário, para grande parte das aplicações. Porém, existem aplicações que precisam ser capazes de decidir por si próprias o que deve ser feito para que seus objetivos sejam atingidos. Este tipo de aplicação é conhecida como agente.

Não existe uma definição universalmente aceita para o termo agente. Em [36] pode-se encontrar uma discussão a esse respeito. Porém, de forma geral, pode-se dizer que agentes são entidades computacionais capazes de perceber e agir sobre um ambiente de forma autônoma [97]. Agentes situados em ambientes complexos, que possuem características como o dinamismo e a imprevisibilidade, por vezes devem ser capazes de agir de forma não apenas autônoma, mas também flexível ou racional. A esse tipo de agente dá-se o nome de agente inteligente. Existem diversos exemplos de aplicações baseadas em agentes inteligentes, como o projeto eSTAR [2], que usa agentes inteligentes para a observação de eventos em estrelas, e os diversos sistemas de busca na Internet baseados em *bots*, *crawlers*, *spiders* e similares [91, 10].

Os três principais conceitos que devem ser compreendidos quando se trata de agentes inteligentes são a autonomia, o ambiente e a flexibilidade [57, 104]. Deve-se notar que, como não há um consenso a respeito destas características, elas devem ser entendidas como linhas gerais, e não como definições rígidas. Elas são descritas abaixo.

A autonomia de um agente é a sua capacidade de agir e tomar decisões sem a intervenção direta de um ser humano ou de outro sistema. Pode-se dizer que um agente deve

possuir controle sobre suas próprias ações e estado interno.

Todo agente deve estar situado em um ambiente. O ambiente no qual um agente se situa é o lugar de onde ele retira informações e no qual ele age. Exemplos de ambientes possíveis para agentes são o mundo real, a Internet e um espaço virtual definido matematicamente em um computador.

A flexibilidade é a característica que determina se um agente é inteligente ou não. Ela é composta por três propriedades: a reatividade, a pró-atividade e a habilidade social. A reatividade é a capacidade do agente de responder a mudanças no ambiente, determinando suas ações de acordo com estas mudanças. Além de responder às mudanças do ambiente, um agente inteligente deve tomar iniciativas, procurando agir de forma a satisfazer os seus objetivos. Essa capacidade de dirigir o seu comportamento é a pró-atividade. Por fim, um agente inteligente deve possuir habilidade social, sendo capaz de usar o ambiente para interagir com outras entidades (pessoas, sistemas ou outros agentes) e tentar satisfazer seus objetivos.

Existem outras características que um agente inteligente pode ou não possuir [57]. Entre estas, podem-se apontar a mobilidade — capacidade de se deslocar dentro do ambiente — e o aprendizado — a capacidade de melhorar o seu desempenho baseado em suas experiências anteriores.

Em [104], pode-se encontrar um modelo que descreve os agentes inteligentes. O ambiente se encontra em um estado, pertencente a um conjunto de estados possíveis. O estado do ambiente pode ser modificado por ações do agente. Um agente possui um estado interno e um conjunto de ações que podem ser executadas. O agente decide pela execução de uma ação a partir do estado do ambiente e do seu estado interno. Em primeiro lugar, ele retira do ambiente informações a respeito do seu estado, transformando-as em um *percept* — a percepção que o agente tem do ambiente. Em seguida, ele atualiza o seu estado interno baseando-se no *percept* e no seu estado interno atual. A partir deste novo estado interno, o agente decide se deve agir ou não e, em caso positivo, escolhe a ação que é executada. A figura 3.1 mostra essa arquitetura.

A partir deste modelo, podem-se descrever algumas das características do ambiente

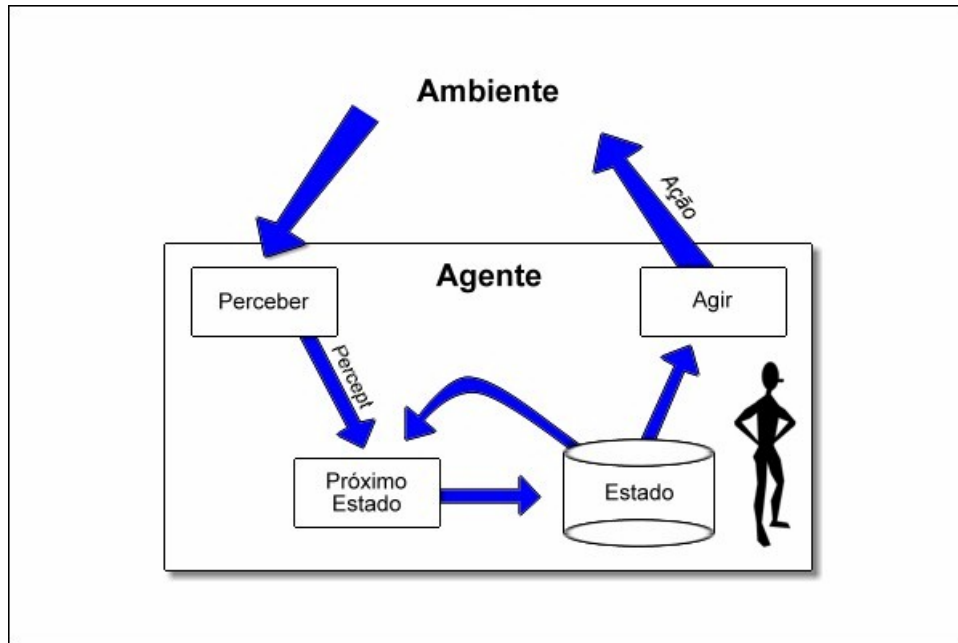


Figura 3.1: **Arquitetura de um agente inteligente.**

[104]. Entre aquelas que podem ser destacadas estão a acessibilidade, o dinamismo e o determinismo. Estas características são descritas abaixo.

Um ambiente acessível é aquele do qual o agente pode ter uma percepção completa e atualizada. Desta forma, para cada estado do ambiente, o agente possui um *percept* distinto. Do contrário, se um mesmo *percept* puder corresponder a dois ou mais estados do ambiente, ele é dito inacessível.

Um ambiente estático é aquele cujo estado só é mudado pelas ações do agente. Já o estado de um ambiente dinâmico pode ser alterado de outras formas, por outras forças ou agentes, o que em geral torna o ambiente imprevisível.

Em um ambiente determinístico, a execução de determinada ação pelo agente quando o ambiente está em certo estado resulta em um único novo estado possível. Já em um ambiente não determinístico, essa ação pode resultar em um conjunto de estados diferentes, o que também torna o ambiente imprevisível.

Pode-se notar que, para todos os efeitos, um ambiente inacessível e dinâmico pode ser considerado não-determinístico. Ambientes complexos costumam ser inacessíveis, dinâmicos e não-determinísticos. Nestes casos, os agentes não são “oniscientes” ou “onipotentes”, possuindo percepção e controle limitados sobre o ambiente. Isto pode tornar o uso de di-

versos agentes, ao invés de apenas um, necessário. Esses sistemas, compostos por diversos agentes, são descritos a seguir.

3.1.1 Sistemas Multiagentes

Um sistema pode ser composto por diversos agentes, que trabalham paralelamente para realizar certo conjunto de tarefas. A esse tipo de sistema, no qual os agentes formam uma “sociedade”, dá-se o nome de sistema multiagente [49]. Existem muitas aplicações baseadas em sistemas multiagentes, tais como sistemas de controle de tráfego aéreo [64], jogos eletrônicos [87] e sistemas de busca na Internet [91, 10].

Sistemas multiagentes podem ser usados em ambientes que são intrinsecamente distribuídos, com estados que são percebidos de forma diferente em lugares diferentes, ou que são complexos demais para aceitar uma solução centralizada. Estes ambientes em geral são aqueles muito extensos (e portanto inacessíveis), muito dinâmicos, ou que possuem muitos componentes. Por isto, cada agente possui sobre o ambiente percepção e controle limitados. Além disso, cada agente é uma entidade autônoma e independente, e não há qualquer controle centralizado sobre os agentes. Essa natureza distribuída e descentralizada dos sistemas multiagentes dá aos mesmos propriedades de robustez e escalabilidade [49].

Para que um sistema multiagente satisfaça seus objetivos, os agentes devem se comunicar e interagir entre si. O ambiente deve prover aos agentes algum meio para que eles usem protocolos para a comunicação e a interação. Como os agentes podem ser heterogêneos e assíncronos, essas interações são muitas vezes imprevisíveis. Isto traz para os sistemas multiagentes várias questões que não existem quando se considera apenas os agentes isolados. De fato, além da inteligência artificial, sistemas multiagentes se beneficiam de estudos em áreas como sociologia, economia, administração, biologia e filosofia [97, 29, 32]. Em [57] são enumeradas algumas das questões que surgem quando se trabalha com sistemas multiagentes:

- Como formular, descrever, decompor e alocar problemas entre um grupo de agentes?

- Quando os agentes devem se comunicar?
- Quais as linguagens e protocolos pelos quais os agentes se comunicam?
- Como equilibrar a computação local e a comunicação?
- Como um agente deve modelar e raciocinar sobre as ações, planos e conhecimento de outros agentes para se coordenar com eles?
- Como reconhecer e conciliar intenções e pontos de vista contraditórios para que um grupo de agentes se coordene?
- Como se certificar de que os agentes estão trabalhando para a satisfação de seus objetivos, sem realizar interações prejudiciais?
- Como saber se o objetivo do sistema já foi satisfeito?
- Quais são as tecnologias e metodologias que podem ser usadas para desenvolver sistemas multiagentes?

Algumas destas questões dizem respeito especificamente à coordenação das ações e comportamentos dos agentes [49, 32]. Seguindo protocolos e políticas de coordenação, os agentes podem otimizar a distribuição de recursos e atividades, determinando objetivos e tarefas em comum, e evitando conflitos desnecessários. Desta forma, os agentes se organizam e melhoram a coerência do sistema, sem que exista para isto um controle centralizado. A coordenação pode ser dividida em cooperação e negociação. A cooperação é a coordenação entre agentes com objetivos similares ou complementares, enquanto a negociação é a coordenação entre agentes que competem entre si ou que possuem objetivos contraditórios. Para que a coordenação seja realizada, cada agente deve possuir um modelo dos outros agentes e das suas possíveis interações, ou seja, habilidades sociais.

Outras questões dizem respeito à forma como os agentes se comunicam entre si. Em sistemas nos quais os agentes podem se mover, a essas questões são somadas diversas outras, relacionadas ao posicionamento dos agentes no ambiente. Algumas destas questões são discutidas a seguir, na seção 3.2, que trata de agentes em redes de computadores.

3.2 Agentes Móveis

Redes de computadores heterogêneas, dinâmicas e de larga escala podem ser vistas como ambientes complexos, nos quais agentes capazes de se deslocar entre os nodos são usados na construção de aplicações distribuídas, flexíveis e tolerantes a falhas [45]. Esta seção discute aspectos relacionados a agentes deste tipo, os agentes móveis. Nas seções 3.2.1 e 3.2.2, são discutidos dois aspectos importantes dos sistemas baseados em agentes móveis: a mobilidade e a comunicação. Na seção 3.2.3, uma arquitetura para um sistema de agentes móveis é apresentada.

Pode-se descrever um agente móvel como um programa autônomo, ou seja, um agente, capaz de se deslocar entre os nodos de uma rede heterogênea. Ele interrompe a sua execução no momento da migração, dando a ela continuidade após a chegada no nodo de destino. Neste trabalho, um agente móvel que apresenta comportamento flexível (reatividade, pró-atividade e habilidades sociais) é chamado de agente móvel inteligente.

A pesquisa na área de agentes móveis tem recebido um grande destaque nos últimos anos, embora possivelmente com um excesso de otimismo. Isto pode ser observado, por exemplo, ao se analisar a transformação que a argumentação de um mesmo grupo de autores sofre em trabalhos como [62, 45, 63, 44]. Nestes trabalhos, o escopo e impacto das aplicações baseadas em agentes móveis vai reduzindo progressivamente, até chegar a uma visão mais concreta e realista. De qualquer forma, é possível encontrar situações nas quais a programação baseada em agentes móveis pode se mostrar vantajosa, como aquelas descritas abaixo.

Nas aplicações em sistemas distribuídos, normalmente um nodo precisa realizar computações que dependem de dados que estão em outros nodos. Este problema pode ser tratado de diversas formas. Por exemplo, os dados podem ser enviados pelos nodos para o nodo interessado, que realiza as computações. Outra possibilidade é uma estratégia na qual os próprios nodos executam processos que realizam as computações, enviando para o nodo interessado apenas os seus resultados. A primeira abordagem pode se mostrar problemática se a quantidade de dados a ser enviada pelos nodos for muito grande. A segunda abordagem pode apresentar problemas no que diz respeito à manutenção de

versões atualizadas dos *softwares* em sistemas com muitos nodos.

Agentes móveis podem ser usados para resolver esses problemas em algumas aplicações [45]. O nodo interessado pode enviar agentes aos nodos que possuem os dados, e as computações são então realizadas localmente pelos agentes. Tendo concluído as computações, eles retornam ao seu nodo de origem, levando os resultados. Essa técnica pode reduzir a utilização da banda da rede, pois muitas vezes o tamanho do código dos agentes é menor que a quantidade de dados que precisaria ser transmitida de um nodo para outro. Nesses casos, podem-se também obter computações mais rápidas, em especial se o enlace entre os nodos é muito lento, já que o tempo gasto na transmissão dos dados é reduzido. Além disso, os agentes encapsulam o código que será executado nos nodos, dispensando atualizações de *software*.

Outros cenários que podem ser beneficiados pelo uso de agentes móveis são aqueles nos quais a conexão de um nodo com a rede não é estável ou é muito limitada. Essa situação é comum, por exemplo, em redes móveis sem fio (apresentadas na seção 2.2). Nestes casos, o nodo pode enviar agentes para a rede e estes podem permanecer em execução por um longo período de tempo, mesmo que o nodo não esteja mais conectado à rede. O enlace precisa existir apenas em dois momentos: quando o nodo envia os agentes para a rede e quando estes agentes são recuperados, junto com os resultados das computações realizadas.

Deve-se observar que em algumas aplicações essas vantagens podem ser obtidas também com o uso de outros tipos de código móvel [39], e não apenas com agentes móveis. Como exemplo de tecnologia para código móvel, podem-se citar os *applets* e *servlets* da linguagem Java [56], pequenos programas que são recebidos e executados por clientes ou servidores na Internet. Outros exemplos são o padrão CORBA para objetos distribuídos [53] e até mesmo a capacidade de atualização automática que vários *softwares* atuais possuem. Desta forma, pode-se afirmar que os agentes móveis são uma técnica útil e de grande potencial, mas que deve ser usada apenas em aplicações que realmente se beneficiem da sua autonomia [63, 44]. O capítulo 4, mais adiante, apresenta uma série de aplicações baseadas em agentes móveis.

Existem diversas questões relacionadas à criação e uso de agentes móveis inteligentes. Nas seções 3.2.1 e 3.2.2 são apresentadas aquelas que são mais relevantes para este trabalho: a mobilidade e a comunicação. As demais questões envolvem principalmente a segurança e a criação de padrões e sistemas para o desenvolvimento de agentes móveis. Estas duas questões são brevemente discutidas a seguir.

Na área de segurança, os principais problemas encontrados são a proteção dos nodos de possíveis agentes maliciosos e dos agentes de possíveis nodos maliciosos [15]. No primeiro caso, as técnicas existentes envolvem sistemas de identificação dos agentes para que os nodos possam reconhecer sua origem. Além disso, procura-se restringir a ação de cada agente a apenas um determinado grupo de nodos ou a certos tipos de operação. No segundo caso, tem-se um problema mais complexo, já que os nodos precisam ter acesso total ao agente para poder executá-lo [1, 59]. Um nodo malicioso poderia alterar informações carregadas pelo agente ou o próprio código do mesmo. Neste caso, são usados esquemas para a encriptação dos dados carregados pelo agente, ou mesmo estratégias baseadas em hardware dedicado [102].

Quanto à padronização, existe atualmente uma grande quantidade de *frameworks* e sistemas que implementam ou provêm um suporte direto à implementação de agentes móveis [46], como o μ Code [83, 82], o NOMADS [30], o Tracy [11] e o D'Agents [23]. Porém, ainda não existe um padrão ou arquitetura que seja universalmente aceito. Existem também diversas questões ligadas à modelagem, validação, testes, depuração e eficiência de sistemas baseados em agentes móveis [54, 33]. A solução de todos esses problemas pode ser bastante complicada em redes descentralizadas, heterogêneas, dinâmicas e de larga escala, e a maior parte deles ainda permanece em aberto [63].

3.2.1 Mobilidade

Existe na literatura um certo consenso quanto à definição mais geral de agente móvel: a de um programa autônomo que pode se deslocar entre os nodos de uma rede heterogênea. Porém, existem diversas abordagens e técnicas que podem ser usadas para que os agentes consigam obter essa mobilidade. Desta forma, há diversas questões que dizem respeito à

mobilidade que devem ser consideradas.

Como agentes móveis são um tipo de código móvel [39], muitas destas questões são comuns àquelas encontradas quando se trabalha com outras abordagens para código móvel. Em [83, 11] pode-se encontrar uma extensa discussão a respeito de muitas delas. Podem-se destacar três aspectos importantes: a autonomia do agente para migrar, a manutenção do estado de execução do agente e a estratégia de migração usada. Esses aspectos são descritos abaixo.

A autonomia do agente para decidir a respeito da sua mobilidade é um dos pontos que devem ser considerados quando se trabalha com agentes móveis. Alguns autores [45, 30, 11] consideram que o agente deve possuir autonomia para decidir quando e para qual nodo se mover. Além disso, essa mobilidade não deve ser ocultada das aplicações que usam os agentes, permanecendo explícita tanto para os agentes quanto para as aplicações. Outros autores [83, 15, 93] assumem que agentes móveis podem ter, além desta mobilidade explícita, uma mobilidade implícita. Neste tipo de mobilidade, as decisões sobre a migração do agente cabem ao nodo no qual o mesmo se encontra, com a movimentação sendo transparente e podendo ocorrer até mesmo sem o conhecimento do agente.

Outra questão relevante diz respeito à manutenção do estado de execução do agente. Pode-se dividir a mobilidade em mobilidade forte e mobilidade fraca. Um agente com mobilidade forte continua a sua execução, após a migração, do ponto onde ela havia sido interrompida. Para isto, além do código do agente e do seu estado interno, o seu estado de execução (contador de programa, pilha de chamadas, etc.) deve ser transmitido. Sistemas como o D’Agents [23] e o NOMADS [30] implementam a mobilidade forte. Já agentes com mobilidade fraca reiniciam sua execução sempre em uma determinada rotina ou “ponto de entrada”, não mantendo o estado de execução após a migração. A maior parte dos sistemas baseados na linguagem Java, como o μ Code [83, 82] e o Tracy [11], implementa a mobilidade fraca.

Por fim, deve-se definir qual é a estratégia que será usada para a migração do código do agente. Em [11, 28], pode-se encontrar uma discussão a respeito de algumas das possíveis estratégias. Aqui são apresentadas as três mais comuns: a *Push-all-to-next*, a *Push-all-*

to-all e a *Pull-per-unit*. Estas estratégias são ilustradas na Figura 3.2. No exemplo da figura, um agente é criado no nodo 0 e migra para os nodos 1, 2 e 3, retornando então ao nodo 0. Em todas as estratégias, o estado interno (e o estado de execução, caso exista mobilidade forte) do agente é enviado do nodo no qual o agente se encontra (o nodo atual) para o nodo para o qual o agente migra (o próximo nodo). O que varia é a forma como o código do agente é transmitido.

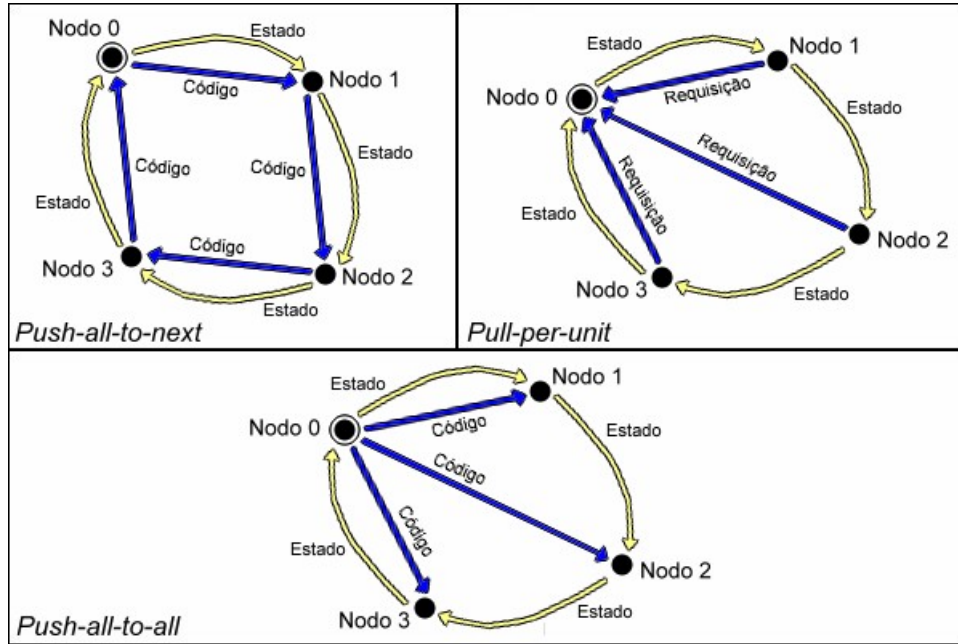


Figura 3.2: Estratégias para o envio do código de um agente.

A estratégia *Push-all-to-next* é a mais tradicional. Nela, o código do agente é transmitido do nodo atual para o próximo nodo no momento da migração, junto com o seu estado. Na estratégia *Push-all-to-all*, o nodo no qual o agente é criado envia o código do agente para todos os nodos pelos quais ele irá passar. Desta forma, apenas o estado do agente precisa ser transmitido no momento da migração, o que pode reduzir o tempo necessário para a execução da sua tarefa. Porém, para que essa estratégia seja possível, o nodo no qual o agente é criado precisa conhecer de antemão todos os nodos pelos quais o agente irá passar e ser capaz de se conectar a estes nodos. Em ambas as estratégias, o código do agente é inteiramente transmitido para os nodos, e muitas vezes os nodos podem não precisar de partes deste código.

Para melhorar a utilização da banda da rede, pode-se optar por estratégias como a Pull-

per-unit. Nesta estratégia, o nodo para o qual o agente migra requisita ao nodo que criou o agente as rotinas ou grupos de rotinas que serão usados. Porém, por exigir um número maior de comunicações, essa abordagem pode aumentar o tempo gasto pelo agente para executar a sua tarefa, em especial se o nodo precisar requisitar as rotinas individualmente ou em grupos pequenos. Além disso, é necessário que todos os nodos sejam capazes de se conectar ao nodo no qual o agente foi criado, o que diminui a autonomia do agente.

Além dos aspectos citados, existem diversos outros que devem ser considerados, como as tecnologias de transmissão usadas e o uso de técnicas que otimizam a utilização da banda da rede, como por exemplo a compactação do código e do estado do agente ou a manutenção de uma cache contendo o código de certos tipos de agente muito comuns.

3.2.2 Comunicação entre Agentes Móveis

Como dito na seção 3.1.1, o ambiente de um sistema multiagente deve prover meios para que os agentes se comuniquem entre si. A comunicação entre agentes é normalmente feita através da troca de mensagens [49]. Em um sistema baseado em agentes móveis, os meios que tornam possível essa troca de mensagens são oferecidos pelos nodos da rede. Deve-se, portanto, definir quais são esses meios e como eles podem ser usados pelos agentes, levando-se em conta a sua mobilidade. Em [15, 13, 14] são discutidas algumas das estratégias usadas por sistemas de agentes móveis para a comunicação. Deve-se notar que um sistema pode implementar mais de uma estratégia, e que outras abordagens diferentes destas são possíveis.

O modelo de comunicação mais comum e intuitivo é a comunicação direta, ilustrada na figura 3.3. Nesse modelo, as mensagens são enviadas diretamente de um agente para outro. A comunicação direta possui algumas características que podem dificultar o seu uso. O agente que envia a mensagem deve conhecer o identificador ou “nome” do receptor, e precisa ser capaz de localizá-lo na rede. Mais que isto, em um sistema onde os agentes podem ser criados e destruídos dinamicamente, é necessário que seja possível determinar, em um dado momento, quais são os possíveis receptores para as mensagens de um agente. Desta forma, além de permitir o envio e recepção de mensagens, o nodo deve oferecer

facilidades para a localização de agentes e roteamento de mensagens. Além disso, no momento em que a mensagem é enviada, o receptor deve ser capaz de recebê-la.

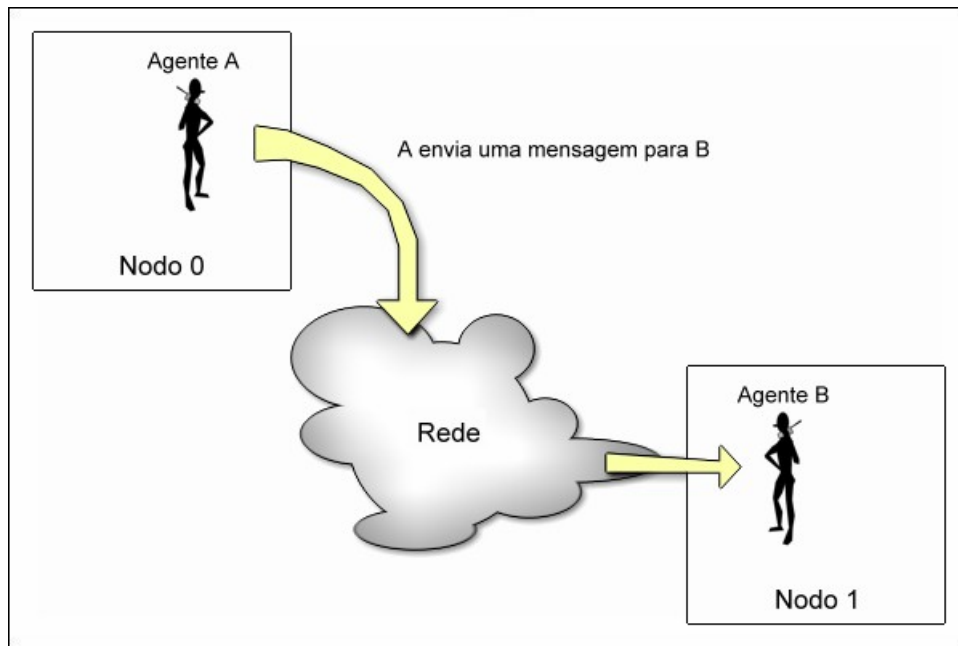


Figura 3.3: **Comunicação direta entre agentes.**

Existem diversas soluções para os problemas da comunicação direta, mas na maior parte dos casos, estas soluções não são simples. Por exemplo, pode-se manter em um nodo conhecido um agente ou servidor para o qual todos os agentes enviam mensagens quando são criados, destruídos ou se movem. Desta forma, o agente ou servidor sabe quais são os agentes existentes e a sua localização. Outros agentes podem então se comunicar com o mesmo antes de enviar uma mensagem, ou mesmo enviar todas as mensagens através dele. Porém, essa solução aumenta a carga na rede com mensagens de controle, acrescenta um possível gargalo no sistema e exige a existência de um nodo estável, conhecido e com o qual todos os agentes possam se conectar. Isto dificulta (se não impossibilita) o uso desta solução em vários casos, como em redes cuja topologia permite particionamentos.

A necessidade de se localizar o receptor pode ser eliminada se a abordagem usada para a comunicação for baseada em “encontros” ou “reuniões”. Nesta abordagem, os agentes trocam mensagens apenas com outros agentes que estejam no mesmo nodo. Para implementar essa estratégia, o nodo precisa prover meios para que os agentes saibam quais outros agentes se encontram localizados nele e para que eles localmente troquem

mensagens entre si. A desvantagem desta abordagem é a dificuldade que um agente pode ter para encontrar um receptor específico, ou mesmo um receptor qualquer para suas mensagens, em especial se o padrão de movimentação dos outros agentes não for conhecido. Isto pode dificultar a coordenação entre os agentes.

Para que os agentes possam todos se encontrar quando for preciso, pode ser necessário, por exemplo, que o nodo e o momento no qual uma reunião irá ocorrer sejam previamente determinados. Porém, em certas ocasiões um agente pode não conseguir chegar ao nodo escolhido no horário agendado e “perder” a reunião — por exemplo, no caso de uma conexão temporariamente perdida, como é o caso da situação ilustrada na figura 3.4. Além disso, é preciso que haja uma sincronização entre os agentes ou os nodos, o que pode ser difícil em ambientes heterogêneos ou de larga escala.

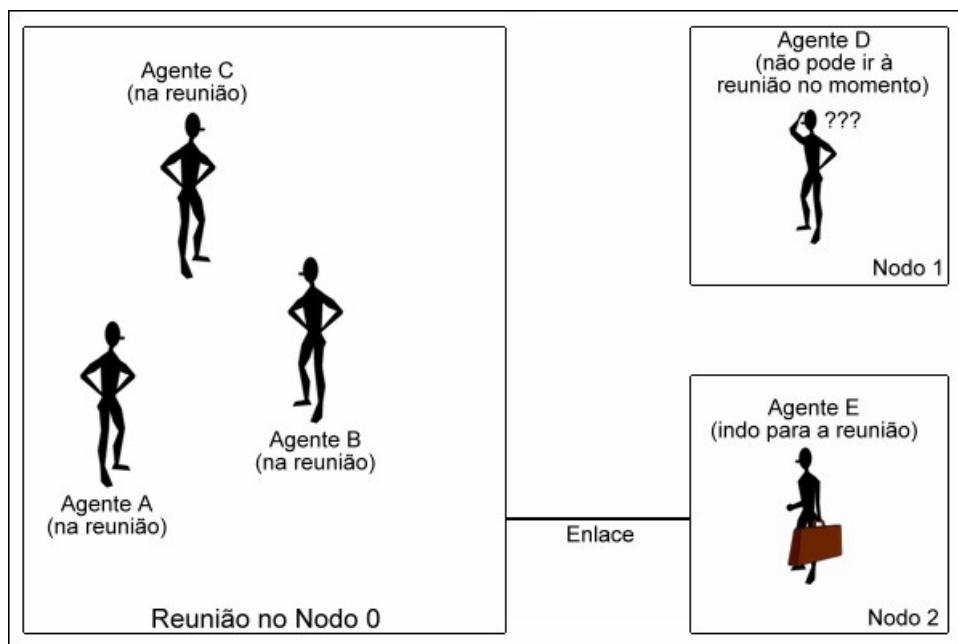


Figura 3.4: Agentes se encontrando para uma reunião.

Pode-se também usar uma abordagem baseada em reuniões “virtuais”, que não se restringem a agentes que estão em um mesmo nodo. Neste caso, um agente inicia a reunião, e outros agentes podem participar da mesma do nodo onde eles se encontram. Para isto, eles devem enviar diretamente suas mensagens para o agente que iniciou a reunião, e este atua como “mediador”, repassando as mensagens para os outros agentes, como mostra a figura 3.5. Uma das desvantagens desta abordagem é a dificuldade que

um agente pode ter para tornar outros agentes cientes do início de uma reunião. Isto pode demandar, por exemplo, algum tipo de mecanismo para a “divulgação” das reuniões iniciadas — o que pode aumentar consideravelmente o tráfego da rede. Além disso, as mesmas dificuldades da comunicação direta podem surgir em certas ocasiões — por exemplo, se um agente se deslocar ou se uma conexão falhar durante a reunião.

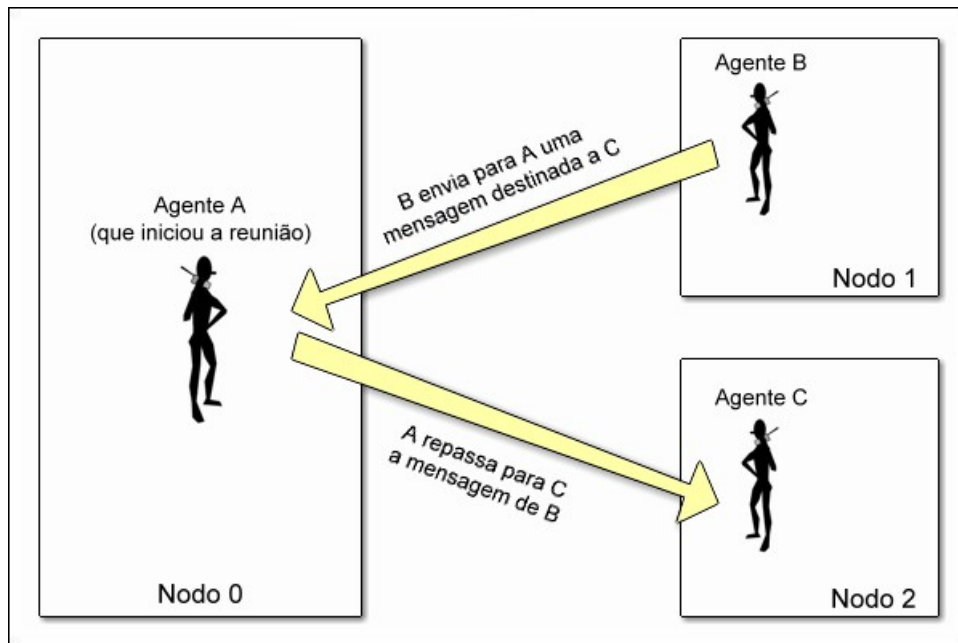


Figura 3.5: Uma reunião com agentes em nodos distintos.

Tanto na abordagem baseada em comunicação direta quanto nas baseadas em reuniões, os agentes trocam mensagens diretamente entre si. Para que os agentes não precisem ser localizados ou sincronizados, pode-se fazer uma troca de mensagens de maneira indireta. Na comunicação indireta, as mensagens não são diretamente enviadas aos seus receptores, permanecendo em um repositório até que possam ser recebidas.

Uma das abordagens para a comunicação indireta é aquela baseada em “quadros de avisos”. Neste modelo, ilustrado na figura 3.6, cada nodo possui um quadro de avisos no qual um agente pode “pendurar” mensagens endereçadas a outros agentes. Quando o agente ao qual a mensagem se destina passar pelo nodo, ela pode ser recuperada. Isto pode ocorrer até mesmo depois que o criador da mensagem tiver sido destruído. Outra abordagem possível é aquela baseada em “quadros negros”. Esta abordagem é similar ao quadro de avisos, com a diferença de que o receptor da mensagem não precisa ser

conhecido. As mensagens podem ser lidas por todos os agentes que passam pelo nodo e recuperadas com base em padrões ou em um cabeçalho comum. Por exemplo, um agente pode deixar no quadro negro uma mensagem destinada a todos os agentes criados em um determinado nodo.

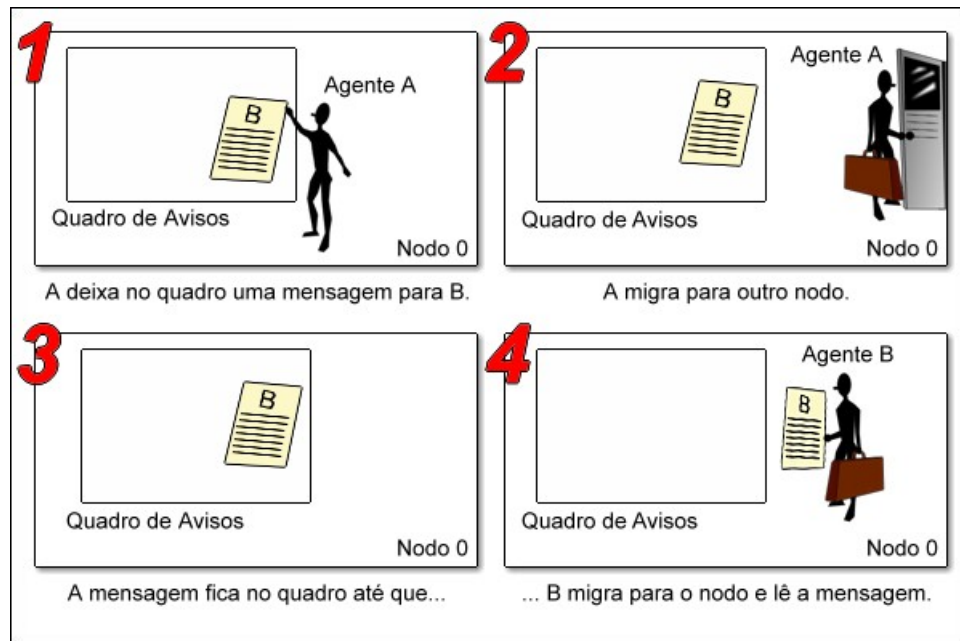


Figura 3.6: **Comunicação por quadro de avisos.**

A comunicação indireta funciona bem em ambientes muito dinâmicos ou nos quais os agentes possuem muita mobilidade [15]. Porém, os repositórios permanecem consumindo recursos dos nodos por um período de tempo possivelmente longo. Além disso, não é possível determinar quando ou se uma mensagem será lida pelos agentes aos quais ela se destina, pois não é possível determinar quando ou se estes agentes irão passar pelo nodo. Isto pode impossibilitar o uso da comunicação indireta em situações nas quais é necessário que uma mensagem seja recebida rapidamente ou instantaneamente, ou mesmo se for importante que a mensagem seja de fato recebida.

As abordagens para comunicação indireta podem também manter um único repositório global. Desta forma, os agentes podem recuperar mensagens sem precisar passar por um nodo específico. Isto pode resolver alguns dos problemas da comunicação indireta, acrescentando a eles, entretanto, outros problemas. É preciso que todos os nodos sejam capazes de se conectar ao repositório global. Além disso, é acrescentado um gargalo ao

sistema. Assim, em certos cenários, abordagens globais para comunicação indireta não são eficientes ou mesmo possíveis. A figura 3.7 mostra um exemplo de quadro negro global, no qual vários agentes contribuem com partes da solução para um mesmo problema.

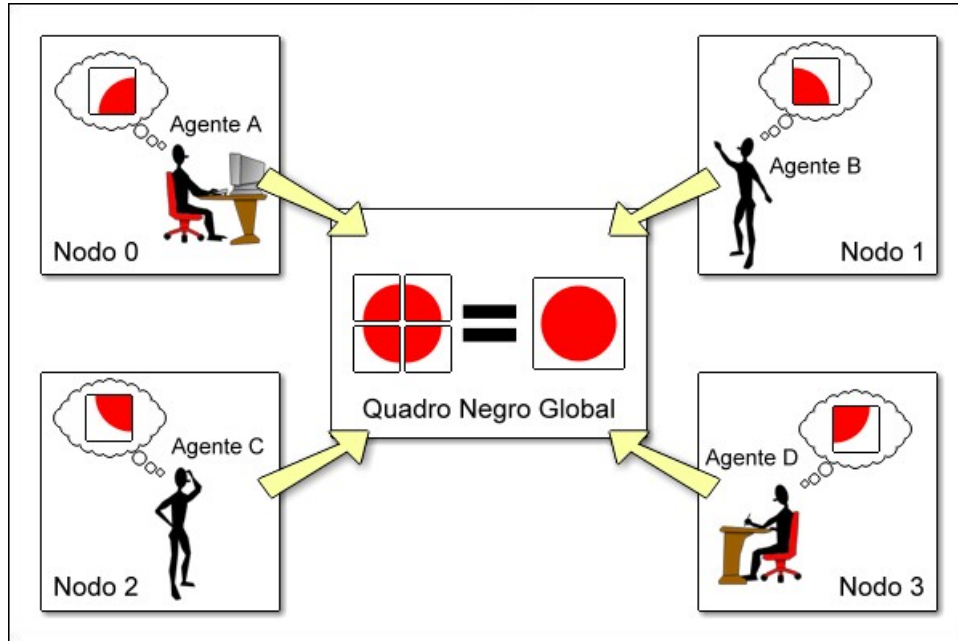


Figura 3.7: Um quadro negro global.

Além das funcionalidades básicas para a comunicação (recepção e envio de mensagens, localização de agentes, repositórios, etc.), os nodos podem possuir capacidades computacionais [13, 14]. Desta forma, eles podem reagir aos acessos feitos pelos agentes e em alguns casos executar tarefas por conta própria. Isto pode ser útil na criação de esquemas de segurança e de protocolos de coordenação entre agentes. Por exemplo, em um sistema baseado em quadros negros, os nodos podem eliminar automaticamente mensagens que fiquem no repositório por um determinado período de tempo. Da mesma forma, cada mensagem pode ter um contador de acessos, que indica quantos agentes já a leram.

Em [15, 13, 14], argumenta-se que em geral a comunicação indireta é a que funciona melhor em sistemas dinâmicos e de larga escala, como a Internet. Porém, pode-se verificar que não existe uma estratégia universal capaz de satisfazer completamente e de forma eficiente todas as aplicações, visto que cada abordagem possui suas próprias vantagens e desvantagens. Além disso, deve-se notar que é preciso definir outros aspectos da comunicação entre agentes, como o formato das mensagens, a maneira como estas mensagens

são interpretadas e como os agentes as usam para se coordenar. Considerações deste tipo também devem ser feitas de acordo com a aplicação, podendo variar de sistema para sistema.

3.2.3 Uma Arquitetura para um Sistema de Agentes Móveis

Nesta seção, é descrita uma arquitetura para um sistema de agentes móveis. Este modelo é baseado naquele oferecido pelo μ Code [83, 82] e na infra-estrutura apresentada em [93]. O modelo considera que cada nodo executa pelo menos um servidor de código móvel. Este servidor permite que o nodo receba conexões de outros nodos, através das quais agentes podem migrar. Ele também disponibiliza para os agentes um ambiente computacional. Este ambiente pode ser dividido em quatro partes: um ambiente de execução, uma interface de transporte, uma interface de comunicação e um conjunto de interfaces de recursos. Estas partes são descritas abaixo. A arquitetura interna de um nodo é mostrada na figura 3.8.

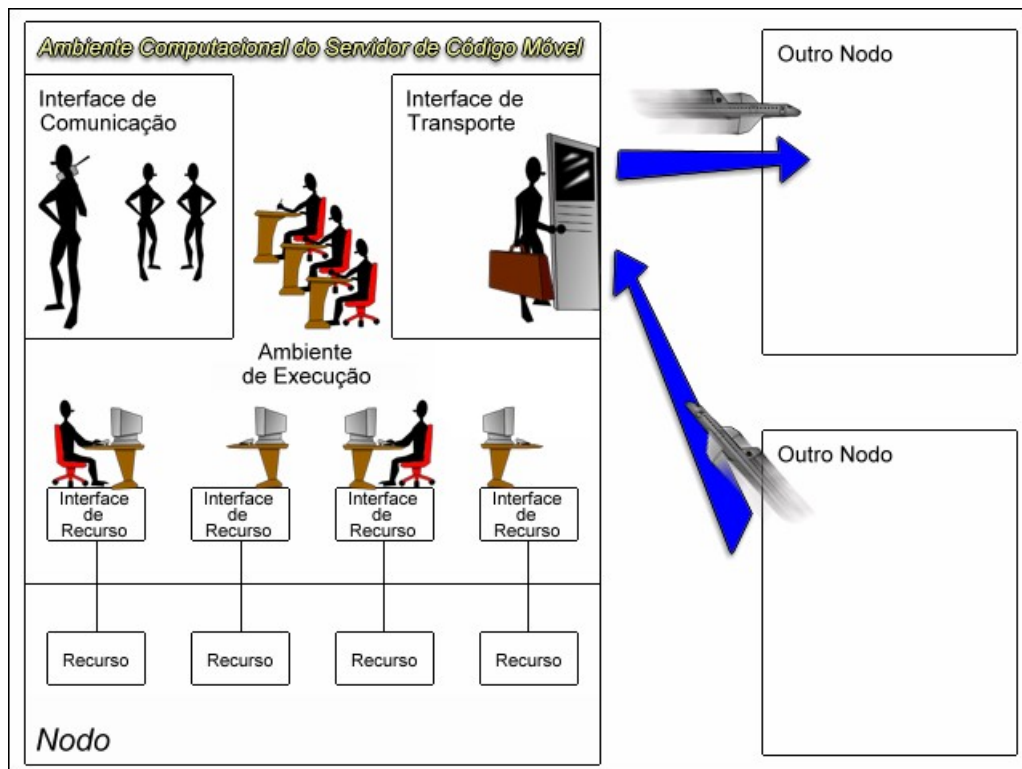


Figura 3.8: Arquitetura interna de um nodo.

O ambiente de execução é a parte do ambiente computacional responsável pela execução do código dos agentes. Para isto, ele possui interpretadores ou compiladores da linguagem na qual os agentes são implementados. Como vários agentes podem estar situados em um mesmo nodo, o ambiente de execução deve organizar os agentes e permitir a sua execução paralela ou escalonada. Desta forma, pode-se dizer que o ambiente de execução é o lugar no qual os agentes “residem” quando estão em um nodo.

A interface de transporte é responsável pela migração dos agentes entre os nodos. Ela permite que sejam transmitidos o código do agente e o seu estado interno (os dados que o agente carrega). Ela oculta dos agentes as camadas inferiores da comunicação e as tecnologias usadas, oferecendo uma interface uniforme e de alto nível para a migração. A interface de transporte mantém a noção de localidade dos agentes, podendo ser acessada por eles, ou seja, existe a mobilidade explícita. Assume-se também que o estado de execução dos agentes não é transmitido. Quando um agente chega em um nodo, sua execução é iniciada em uma rotina conhecida, ou seja, a mobilidade é fraca. Além disso, o modelo assume a estratégia *Push-all-to-next*, na qual todo o código do agente é transmitido de um nodo para o outro no momento da migração. Pode-se dizer que a interface de transporte é por onde os agentes “entram” e “saem” de um nodo, e é o meio usado por eles para “viajar” entre nodos.

A interface de comunicação deve prover aos agentes meios para que eles troquem mensagens entre si. Essa parte do sistema implementa esquemas de comunicação como aqueles apresentados na seção 3.2.2. Como cada abordagem possui vantagens e desvantagens próprias, a interface de comunicação não assume o uso de uma única estratégia, deixando em aberto a escolha de uma ou outra de acordo com as necessidades de cada aplicação. Pode-se dizer que a interface de comunicação oferece aos agentes os meios para que eles “conversem” entre si.

Cada interface de recurso oferece aos agentes uma forma padronizada e segura de acesso a um recurso presente no nodo. Através desta interface, os agentes podem acessar de uma mesma maneira recursos similares que tenham formas de acesso distintas (por exemplo, equipamentos com um mesmo propósito produzidos por fabricantes diferentes).

Exemplos de recursos são impressoras, arquivos, placas de som e espaço em disco. Cada nodo possui um conjunto de interfaces de recursos. Pode-se dizer que uma interface de recurso é uma ferramenta usada pelo agente para “interagir com o ambiente”.

Deve-se notar que, além dos agentes móveis, essa arquitetura é capaz de comportar agentes estáticos. Estes são simplesmente agentes que não migram, permanecendo em um mesmo nodo. Afora isto, eles funcionam da mesma forma que os agentes móveis, “residindo” no ambiente de execução, “conversando” através da interface de comunicação e “interagindo com o ambiente” através de interfaces de recurso.

Além das partes descritas, o ambiente computacional pode oferecer facilidades para a criação e destruição de agentes por parte do nodo ou de outros agentes. Além disto, diversos esquemas de segurança podem ser implementados, baseando-se em técnicas tais como a encriptação de dados, restrições de acesso e a aplicação de políticas de execução. Um exemplo de esquema de segurança é um sistema de identificação e autenticação que indica se o agente é confiável ou não, para que se decida se ele deve ou não ser executado. Outro esquema de segurança poderia manter restrito o acesso a certos recursos do nodo para certos agentes. Mais um exemplo é o uso de um agente estático capaz de detectar agentes maliciosos e requisitar a sua destruição. Deve-se observar que tais esquemas de segurança não são obrigatórios. Por exemplo, em uma implementação experimental desta arquitetura, todos os agentes podem ser conhecidos ou terem sido criados por um mesmo grupo, o que torna alguns esquemas de segurança desnecessários.

3.3 Inteligência Coletiva e Sistemas Baseados em Colônias de Formigas

Agentes móveis localizados em redes dinâmicas e descentralizadas, como aqueles considerados neste trabalho, podem encontrar situações que dificilmente poderiam ter sido previstas no momento da sua criação. Para que eles possam se coordenar e cumprir suas tarefas com algum grau de otimização, torna-se interessante que eles tenham algum tipo de inteligência. Como exemplo disso, pode-se citar [28], onde é discutido o papel da

inteligência dos agentes na escolha entre diversas estratégias de migração.

Deve-se ressaltar que um agente móvel não precisa necessariamente ser inteligente: suas decisões podem não ser flexíveis, podendo até mesmo ser tomadas por outra entidade, como o nodo que o criou ou o nodo no qual ele se encontra. Além disso, a flexibilidade de um agente móvel inteligente pode não ter qualquer influência sobre a sua mobilidade, podendo estar relacionada apenas a outras tarefas que o agente executa quando está em um nodo.

Nesta seção, é apresentado um modelo específico para uma sociedade de agentes móveis inteligentes — os sistemas baseados em colônias de formigas. Estes sistemas se fundamentam em diversos princípios da chamada “vida artificial” [78], mais especificamente na *swarm intelligence* [9, 61] — termo que pode ser traduzido como “inteligência de enxames” ou, de forma mais livre, “inteligência coletiva”. Esta segunda tradução é aquela usada neste trabalho. A inteligência coletiva é descrita na seção 3.3.1, enquanto a seção 3.3.2 destina-se à metáfora da colônia de formigas propriamente dita.

3.3.1 Inteligência Coletiva

A inteligência coletiva é uma propriedade de sistemas nos quais um grupo de agentes simples exhibe coletivamente um comportamento complexo. As pesquisas na área de inteligência coletiva procuram desenvolver algoritmos para a solução distribuída de problemas, de uma forma inspirada principalmente no comportamento de colônias de insetos sociais, como formigas, abelhas e cupins — colônias estas que possuem uma versão natural de inteligência coletiva.

Por exemplo, em sua jornada para a construção de um formigueiro, grupos de formigas tecelãs são capazes de formar verdadeiras “pontes vivas” entre folhas, através das quais outras formigas da colônia podem passar. Por vezes, estas pontes exercem uma força suficiente para unir as duas folhas; e nestes casos as suas extremidades são grudadas com a seda produzida por larvas que são carregadas por formigas trabalhadoras, permitindo que as formigas que formavam a ponte executem outras ações [48].

Em uma colônia deste tipo, existem diversas tarefas complexas que devem ser reali-

zadas, como a coleta de alimentos e a construção de ninhos. Cada membro da colônia possui alguma adaptabilidade e flexibilidade, mas raramente é capaz de realizar sozinho uma destas tarefas. As tarefas são então realizadas em conjunto, com o trabalho sendo dividido entre os membros da colônia de acordo com certos mecanismos ou estruturas de castas, baseado-se em critérios como a idade ou uma anatomia especializada para certa função. Essa divisão de tarefas ocorre sem que exista um controle centralizado ou um planejamento prévio, e cada membro age de maneira autônoma.

Desta forma, a colônia pode ser vista como um “superorganismo”, um sistema vivo cuja capacidade de resolver problemas complexos e se auto-organizar é fruto da sinergia e das interações que ocorrem entre os insetos que a formam [48]. Desta observação, surge a idéia de que existem situações nas quais um comportamento complexo “emerge” da união de vários comportamentos mais simples [78]. Ou seja, a união de várias inteligências “menores” pode produzir uma inteligência “maior”.

Pode-se observar que essas colônias de insetos sociais são capazes de realizar de maneira distribuída e descentralizada conjuntos de tarefas em um ambiente complexo — o mundo real. Elas são sociedades formadas por entidades autônomas — os insetos. É possível então traçar um paralelo entre essas colônias e sistemas multiagentes. E, desta forma, elas podem servir como inspiração para a busca por maneiras pelas quais o comportamento de agentes simples pode ser coordenado para que um sistema multiagentes apresente a inteligência coletiva [9, 61]. A idéia de inteligência coletiva em sistemas multiagentes pode ser usada em aplicações em áreas diversas, como o controle de congestionamentos em sistemas de telecomunicações [96, 98] e a criação de efeitos especiais para o cinema [66].

3.3.2 Sistemas Baseados em Colônias de Formigas

Entre os sistemas computacionais que procuram apresentar a inteligência coletiva, podem-se destacar aqueles baseados em colônias de formigas [26, 47, 101]. Nesses sistemas, os agentes se comunicam através de um esquema encontrado na natureza em seres como cupins e formigas — a estigmergia [42, 103, 48]. Esta é uma forma de comunicação na

qual as ações de um inseto sobre o ambiente podem afetar as ações de outros insetos. Um inseto pode não ter sequer conhecimento da presença ou mesmo da existência dos demais, percebendo apenas as modificações feitas por eles no ambiente. Portanto, a comunicação se dá de maneira indireta e localizada. Pode-se dividir a estigmergia em duas categorias [101], descritas abaixo: a comunicação sematectônica e a estigmergia baseada em sinais.

A comunicação sematectônica [42, 103] ocorre quando os insetos se comunicam através da observação do progresso da própria tarefa que está sendo realizada por eles. Por exemplo, ao participar da construção de um formigueiro, uma formiga percebe o estado atual da construção e acrescenta, se necessário, sua contribuição. A construção não segue um plano inicial ou global, avançando à medida em que formigas sentem a necessidade de continuá-la.

Na estigmergia baseada em sinais [48], um inseto modifica o ambiente de uma forma que não influi diretamente sobre o progresso da tarefa executada, mas que afeta o comportamento de outros insetos. Em geral, essas alterações possuem duração limitada e são feitas através de substâncias que são depositadas no ambiente. Por exemplo, quando procura por uma fonte de alimento, uma formiga se desloca de maneira praticamente aleatória. Ao encontrar uma destas fontes, ela retorna ao formigueiro, carregando uma certa quantidade de alimento e deixando uma trilha de feromônio — um hormônio altamente volátil. Essa trilha marca o caminho entre o formigueiro e a fonte de alimento. Outra formiga que esteja procurando por alimento, ao encontrar uma trilha deste tipo, tem uma certa probabilidade de segui-la.

Caso isto ocorra, ela também poderá encontrar a fonte de alimento, e reforçará a trilha ao retornar para o formigueiro — o que torna mais provável que outras formigas a sigam. Desta forma, enquanto houver alimento na fonte, formigas deverão seguir e reforçar a trilha. Ainda assim, a chance de uma formiga seguir ou não uma trilha é probabilística, e a exploração de caminhos alternativos é possível. Como o feromônio é altamente volátil, se uma formiga encontrar um caminho mais curto, sua trilha deverá ter uma força maior, e existe uma tendência de que esta passe a ser seguida no lugar da anterior. Quando a fonte de alimento se esgota, a trilha deixa de ser reforçada e, conforme o feromônio

evapora, a fonte é “esquecida” pelas formigas.

Um agente em um sistema baseado em colônias de formigas possui capacidade limitada, pouca ou nenhuma memória, e um certo componente estocástico em seu comportamento. Ele também deve ser capaz de depositar e detectar no ambiente “feromônios”, ou algo que desempenhe papel equivalente. Podem existir vários tipos de feromônio, e um agente pode ter diferentes graus de sensibilidade a cada um deles. Além disso, os feromônios podem reagir entre si de diversas formas. O agente não possui um modelo explícito dos outros agentes, tendo suas habilidades sociais implicitamente codificadas em sua maneira de alterar e perceber o ambiente.

Como exemplos de sistemas multiagentes que se baseiam na comunicação sematectônica, podem-se citar diversos experimentos na área da robótica nos quais os agentes procuram cooperar para construir estruturas [27, 80, 47], de uma forma inspirada na construção de formigueiros. Já a estigmergia baseada em sinais pode ser usada como base para algoritmos de otimização que têm como inspiração a procura de alimentos pelas formigas. Estes algoritmos servem para solucionar problemas estáticos, como o do caixeiro viajante [21, 26], ou dinâmicos, como o roteamento [16, 96, 98], a distribuição de cargas [90], e a detecção de falhas [101] em redes de telecomunicações.

No caso de agentes móveis, como os considerados neste trabalho, a comunicação sematectônica é possível se as tarefas realizadas por um agente em um nodo afetarem o ambiente de uma forma que possa ser percebida pelos demais agentes. Já a estigmergia baseada em sinais pode ser usada se os nodos possuírem um local no qual os agentes possam “depositar” feromônios. Os dois tipos de estigmergia podem ser obtidos se a comunicação entre agentes se der através de um sistema semelhante (ou mesmo idêntico) à comunicação baseada em quadros negros, apresentada na seção 3.2.2.

A estigmergia pode ser usada pelos agentes móveis para tomar decisões a respeito da migração e da comunicação. Por exemplo, se os nodos permitirem que os feromônios sejam associados a um determinado enlace, um agente pode usar essa informação, associada a outros critérios, para decidir quando e para qual nodo migrar. Outro exemplo seria a liberação de um feromônio pelo agente quando este percebe que a execução de uma

tarefa chegou a um estágio determinado, estimulando uma mudança no comportamento de outros agentes.

De acordo com as necessidades de cada aplicação, esse modelo pode ser adaptado de diversas formas. Por exemplo, a metáfora da colônia de formigas pode ser usada apenas no que diz respeito à migração, com os agentes possuindo habilidades de comunicação e processamento mais complexas para realizar outras atividades. Outro exemplo seria um aumento na capacidade de memória dos agentes, o que possibilitaria o armazenamento de informações tais como dados necessários para a execução de suas tarefas ou os nomes de nodos já visitados.

Capítulo 4

Trabalhos Relacionados

Neste capítulo, são apresentadas as idéias nas quais se fundamentam diversos trabalhos relacionados. A relação entre os trabalhos descritos e este está no uso de agentes móveis em aplicações similares à descoberta da topologia, ou que têm como ambiente redes dinâmicas como aquelas apresentadas no capítulo 2. Além disso, muitos deles se baseiam na metáfora da colônia de formigas, apresentada na seção 3.3. As idéias desses trabalhos servirão como ponto de partida para a proposta aqui apresentada. Os trabalhos são descritos apenas em linhas gerais, com as idéias mais relevantes para a estratégia proposta neste trabalho sendo detalhadas mais adiante, no capítulo 5.

O restante deste capítulo se divide da seguinte forma. Inicialmente, na seção 4.1, são descritas estratégias para duas aplicações relacionadas ao gerenciamento de redes de maneira geral: a modelagem e a localização de falhas. Nas seções 4.2 e 4.3, são apresentadas soluções, respectivamente, para o roteamento e para a localização de recursos em redes P2P. Por fim, na seção 4.4 é descrita uma estratégia na qual, de maneira similar àquela proposta por este trabalho, agentes móveis são usados para a descoberta e manutenção, nos nodos, de uma descrição da topologia de uma rede dinâmica.

4.1 Agentes Móveis para o Gerenciamento de Redes

Agentes móveis podem ser usados de diversas formas em aplicações para o gerenciamento de redes [7]. Tais aplicações incluem a medição de parâmetros de desempenho, a configu-

ração automática de serviços e recursos, ou mesmo a construção de redes *plug-and-play*. Nesta seção, são apresentadas algumas idéias a respeito do uso de agentes móveis em duas aplicações: a modelagem de redes e a localização de falhas. Em ambos os casos, assume-se a existência de um nodo central, a estação de gerência, que é de onde os agentes partem e para onde eles devem enviar os seus resultados.

A modelagem de redes [7, 100] envolve a descoberta de vários aspectos relacionados aos componentes de uma rede. No caso mais simples, esta descoberta refere-se apenas aos nodos que fazem parte da rede. Modelos mais elaborados podem incluir os enlaces (o que iguala a modelagem à descoberta da topologia); recursos ou serviços oferecidos; ou mesmo o estado de componentes. A descoberta pode ser feita por agentes que visitam os nodos e enviam à estação de gerência mensagens contendo informações sobre cada nodo visitado.

A modelagem pode ser feita de forma esporádica (por exemplo, quando um usuário emite um comando na estação de gerência) ou contínua. Um processo de descoberta contínuo pode se mostrar útil em redes dinâmicas, e pode ser obtido com o uso de um agente que esteja sempre migrando entre os nodos e identificando componentes. Essa tarefa pode ser realizada por vários agentes, com um aumento no seu número podendo resultar em uma atualização mais rápida do modelo — mas acompanhando um aumento na carga da rede. Outra forma de se tornar a descoberta constante é com a criação de agentes que permanecem nos nodos, avisando à estação de gerência das mudanças que ocorrem nos mesmos.

O padrão de migração dos agentes pode ser determinado pela estação de gerência a partir de um modelo anterior da rede, ou pelos próprios agentes durante o processo de descoberta. A primeira abordagem permite um melhor aproveitamento dos recursos da rede e o uso de agentes mais simples. Porém, como o propósito dos agentes é exatamente criar um modelo da rede, essa estratégia pode não ser possível. Deixar as decisões a respeito da migração a cargo dos agentes exige o uso de algoritmos eficientes e, possivelmente, de protocolos para a coordenação.

Em [7], afirma-se que apenas a descoberta da topologia da rede não justificaria o uso de

agentes móveis, por conta da complexidade que é acrescentada ao sistema. Porém, aquele trabalho trata de redes com pouco dinamismo, e considera a existência de uma estação de gerência centralizada. Portanto, tal afirmação não pode ser estendida à descoberta da topologia em redes dinâmicas e descentralizadas nas quais cada nodo deve ter uma descrição da topologia da rede.

Baseando-se em princípios similares aos da modelagem, podem-se usar agentes móveis para a localização de falhas em nodos [7]. Por exemplo, se os agentes forem capazes de realizar medições, pode-se limitar a descoberta a componentes que tenham utilização superior a certo limiar; o que permite a localização de gargalos e de recursos sobreutilizados. Podem-se também dividir os agentes em classes com responsabilidades diferenciadas. Por exemplo, agentes de um certo tipo podem ser usados para detectar anomalias e avisar a estação de gerência da sua existência. A estação de gerência pode então criar outros agentes, para estudar o problema mais a fundo ou mesmo tentar corrigir as falhas encontradas.

A idéia da divisão dos agentes em várias classes também pode ser encontrada em [99]; trabalho que também usa a metáfora da colônia de formigas. Os agentes são responsáveis pelas decisões a respeito da migração, e se comunicam através de “mensagens químicas”, que são depositadas nos nodos. Alguns agentes executam medições nos nodos e deixam no ambiente mensagens que indicam alterações em componentes. Estas mensagens são usadas para direcionar outros agentes para os locais da rede que exigem mais atenção.

Outras propostas para o uso de agentes móveis para o gerenciamento de redes incluem as arquiteturas apresentadas em [41, 85, 89]. Nestes sistemas, a idéia central também é a divisão dos agentes em classes com funções distintas, como a coleta de dados de um certo tipo; a detecção de falhas ou eventos específicos; ou mesmo a criação de itinerários para outros agentes.

4.2 Agentes Móveis para o Roteamento

Algoritmos para o roteamento em redes de comunicação figuram entre as primeiras aplicações dos sistemas baseados em colônias de formigas [16, 96, 98]. Esses algoritmos usam agentes para a construção de tabelas de roteamento. Cada nodo possui uma tabela, que

pode conter rotas completas entre os nodos, ou apenas o próximo nodo destas rotas — o nodo para o qual o tráfego deve ser direcionado de forma a chegar a um destino.

Os primeiros algoritmos se inspiram diretamente na procura de alimento pelas formigas [16, 98]. Para descobrir a rota entre dois nodos, são criados vários agentes, que partem da origem e migram entre os nodos da rede procurando pelo destino. Um agente que encontre o destino retorna ao nodo de origem, fazendo o caminho inverso e marcando a rota descoberta através de “feromônios” ou de atualizações nas tabelas de roteamento dos nodos. As rotas descobertas afetam probabilisticamente o itinerário dos agentes. Inicialmente, os nodos para os quais os agentes migram são escolhidos aleatoriamente, mas conforme rotas são descobertas, os agentes tendem a segui-las. Como ocorre na busca de alimentos pelas formigas, existe a possibilidade de rotas alternativas serem descobertas e uma tendência de que as rotas mais curtas sejam seguidas por mais agentes.

Uma abordagem similar a essa pode ser usada para o balanceamento da carga em redes de telecomunicações [90]. Neste caso, é feito um roteamento dinâmico, no qual se procura evitar congestionamentos com a escolha de nodos cuja carga no momento é menor. A carga de cada nodo pode variar, mas considera-se que alterações na topologia da rede raramente ocorrem. Esse algoritmo procura melhorar o desempenho dos anteriores ao usar a “idade” dos agentes como parâmetro para a atualização das tabelas de roteamento. Esta idade é dada em função do número de enlaces pelos quais o agentes já passou, assim como o grau de congestionamento dos mesmos.

Uma proposta diferente pode ser encontrada em [4, 3]. Nesses trabalhos, é apresentada uma versão baseada em agentes para o algoritmo de roteamento de vetor de distâncias (*Distance Vector Routing* - DVR) [69], o *Agent-Based* DVR (ADVR). No ADVR, os agentes não possuem origem ou destino determinados, ficando na rede constantemente enquanto carregam a tabela de roteamento de cada nodo visitado para um dos seus vizinhos. Desta forma, os agentes servem basicamente como disseminadores da informação encontrada nos nodos.

A eficiência do ADVR está ligada à capacidade que os agentes têm de se “espalhar pela rede” da melhor forma possível. Se todos os agentes seguirem caminhos similares, pouca

ou nenhuma informação será levada aos nodos visitados. Além disso, alguns nodos podem nunca receber as tabelas de alguns dos seus vizinhos. Para que os agentes se espalhem pela rede de uma forma eficiente, devem-se usar boas estratégias de migração. Uma possível estratégia é a escolha aleatória entre os vizinhos do nodo atual. Esta estratégia pode ser melhorada se a escolha for feita com base em algum critério, como o número de visitas feitas a cada vizinho ou o número de vezes que um enlace foi usado. Outra estratégia faz o uso de feromônios, que neste caso têm um funcionamento inverso do convencional, diminuindo a probabilidade de um agente seguir o mesmo caminho que outro agente fez há pouco tempo.

Agentes móveis também podem ser usados em algoritmos para o roteamento em redes móveis sem fio ad hoc, apresentadas na seção 2.2. Por exemplo, [5] propõe a combinação de estratégias como aquelas apresentadas a técnicas para a conservação de energia. Já [73] apresenta uma estratégia similar ao ADVF, na qual cada agente possui um histórico dos enlaces através dos quais ele migrou. Este histórico é usado para atualizar as tabelas de roteamento dos nodos. Estratégias como essa também podem ser combinadas com algoritmos para a descoberta de rotas sob demanda [70]. Diferente dos algoritmos que mantêm tabelas de roteamento, esse tipo de algoritmo descobre as rotas apenas quando as mesmas se fazem necessárias. Observa-se que, em estratégias como essas, parâmetros como o tamanho do histórico e da população de agentes exercem grande influência no desempenho do algoritmo e na carga gerada sobre a rede.

4.3 Agentes Móveis para a Localização de Recursos em Redes P2P

Para poder requisitar um recurso, um nodo em uma rede P2P (ver seção 2.1) precisa ser capaz de determinar quais nodos o possuem. Redes centralizadas podem manter índices associando os recursos aos nodos nos quais eles podem ser encontrados. Redes descentralizadas estruturadas podem manter este tipo de índice em “supernodos”, ou posicionar os recursos em nodos específicos [92, 107, 86]. Em redes descentralizadas não estruturadas,

tais abordagens não são possíveis. Nesse tipo de rede, o algoritmo mais simples que pode ser usado é o da inundação [67]: o nodo inicia a busca enviando uma mensagem para os seus vizinhos, e cada nodo que recebe a mensagem a repassa para os seus próprios vizinhos. Quando um nodo que possui o recurso procurado recebe a mensagem de busca, ele envia uma resposta ao nodo interessado, que pode então fazer a requisição. Essa solução não é escalável, e pode sobrecarregar a rede com mensagens de busca.

Outras abordagens procuram contornar esse problema [67, 105, 84]. Em geral, essas técnicas aumentam o tempo necessário para se localizar um recurso, mas reduzem consideravelmente o número de mensagens transmitidas. Pode-se destacar a estratégia baseada em caminhadas aleatórias ou disseminação de rumores. Nesta abordagem, o nodo envia a mensagem de busca para um único vizinho, escolhido aleatoriamente. Se o recurso não for encontrado no nodo que recebeu a mensagem, ela é repassada para outro vizinho, também escolhido aleatoriamente, com o processo se repetindo até que o recurso seja encontrado ou até que a mensagem tenha visitado um determinado número de nodos. Para melhorar o desempenho desse algoritmo, podem-se usar várias mensagens, ao invés de apenas uma. Além disso, os nodos para os quais as mensagens são enviadas podem ser escolhidos com base em algum critério, ao invés de aleatoriamente.

Essa estratégia pode ser estendida, com as mensagens de busca sendo encapsuladas em agentes móveis. Estes passam então a tomar, no lugar dos nodos, as decisões a respeito de como as mensagens devem ser disseminadas. Esta idéia pode ser observada, por exemplo, em [24]. Na abordagem proposta por esse trabalho, um nodo pode criar agentes móveis que migram entre os nodos da rede em busca de um determinado recurso. Se um agente localizar o recurso, ele retorna ao nodo de origem, levando consigo informações a respeito do itinerário percorrido e do nodo no qual o recurso foi encontrado. Estas informações são usadas para construir uma “trilha”, que pode ser usada para que agentes criados futuramente encontrem recursos com mais rapidez.

Outra abordagem para a localização de recursos que se baseia em agentes móveis pode ser encontrada em [106]. Nesta estratégia, que se inspira na metáfora da colônia de formigas, cada nodo possui um pequeno índice com informações sobre os recursos

compartilhados em outros nodos. Estes índices são mantidos pelos agentes, e usados por eles para a localização de recursos. Quando um recurso se torna disponível em um nodo, um agente é criado e visita um certo número de nodos, atualizando os seus índices. Para procurar por um recurso, um nodo cria agentes de outro tipo, que usam os índices para tomar decisões sobre a migração. Esses agentes também atualizam os índices conforme migram entre os nodos, melhorando o desempenho da busca através da comunicação sematectônica.

4.4 A Estratégia de RoyChoudhury, Paul e Bandyopadhyay para a Descoberta da Topologia

Em [20], é descrita uma estratégia que usa agentes móveis para a descoberta da topologia de uma rede móvel sem fio ad hoc. Os agentes migram entre os nodos, mantendo nos mesmos uma descrição da topologia da rede, de maneira distribuída e descentralizada. Essa estratégia, descrita nesta seção, é em vários aspectos similar à proposta deste trabalho.

Cada nodo possui uma descrição da topologia da rede, que é diretamente acessada pelos agentes. Além de poder ser usada por aplicações executadas no nodo, esta descrição é usada como meio de comunicação pelos agentes — de maneira similar a um quadro negro (descrito na seção 3.2.2). Os agentes carregam entre os nodos informações obtidas nessas descrições. Assume-se que um nodo sempre sabe quais são os seus vizinhos, mas esta é a única informação diretamente disponível. Informações sobre o restante da rede devem ser levadas ao nodo por agentes.

Um agente, ao chegar a um nodo, atualiza a descrição da topologia com informações trazidas de outros nodos. Antes de migrar para outro nodo, o agente permanece inativo por um certo tempo. Esse tempo antes de migrar serve para diminuir o tráfego de agentes na rede. Enquanto o agente está inativo, a descrição da topologia pode mudar, por alterações nos vizinhos do nodo ou pela visita de outros agentes. Portanto, após o término do tempo antes de migrar, o agente atualiza as suas informações com aquelas encontradas na descrição da topologia, migrando em seguida para um dos vizinhos do nodo.

Dado o dinamismo da rede, mudanças podem ocorrer entre o instante em que uma informação é obtida por um agente e a sua chegada em um nodo. Desta forma, a informação pode se tornar desatualizada, não mais correspondendo à realidade. Além disso, dois agentes podem chegar a um mesmo nodo em instantes diferentes carregando informações contraditórias a respeito de certo nodo ou enlace. Por isto, agentes e nodos devem procurar sempre usar as informações mais recentes disponíveis. Para determinar a “idade” de uma informação, cada nodo mantém um contador, que é iniciado em 0 e incrementado quando um agente migra do nodo para um dos seus vizinhos. Todas as informações obtidas a respeito do nodo são associadas a este contador. Portanto, quanto maior é este contador, mais recente — e provavelmente mais próxima da realidade — é a informação.

O dinamismo da rede também exerce influência sobre as informações que estão na descrição da topologia em um nodo. Se não forem atualizadas, com o passar do tempo estas informações tendem a se tornar cada vez mais distantes da realidade. Por isto, é definido o conceito de envelhecimento da informação. A cada instante de tempo, as informações presentes em um nodo se tornam menos confiáveis. Desta forma, se uma informação permanece por muito tempo em um nodo sem ser atualizada, sua confiabilidade pode se tornar baixa demais para ser aceita — o que pode levar ao seu descarte.

Assim como ocorre no ADVR (descrito na seção 4.2), nessa estratégia os agentes são usados para disseminar informações entre os nodos. Portanto, os agentes devem se espalhar da melhor forma possível pela rede. Para que isto ocorra, o próximo nodo a ser visitado por um agente é definido como sendo o vizinho do nodo atual que recebeu menos visitas de agentes no total — exceto em casos especiais, nos quais o mesmo vizinho já foi visitado várias vezes por agentes que partiram do nodo. Isso pode ser determinado a partir do mesmo contador que é usado para definir a idade das informações.

Também é necessário determinar quantos agentes são usados, ou o tamanho da população. Se esta população é muito pequena, as informações levam muito tempo para serem levadas aos nodos, e as descrições da topologia tendem a ser pouco confiáveis. Se a população é muito grande, um grande número de agentes visita cada nodo em um período de tempo pequeno demais para que a topologia da rede tenha sofrido alterações — e pouca

ou nenhuma informação relevante é obtida. Desta forma, recursos são consumidos sem que o desempenho do sistema melhore. Experimentos indicam que existe um tamanho máximo ótimo para a população de agentes, a partir do qual pouca ou nenhuma melhora pode ser observada na descrição da topologia nos nodos. Este tamanho varia de rede para rede, de acordo com o dinamismo, a distribuição e o número de nodos.

Essa estratégia é, em vários aspectos, bastante similar àquela proposta neste trabalho. Entretanto, ela tem algumas limitações. Ela considera que todos os nodos da rede, assim como os seus contadores, são iniciados em um mesmo momento, e os algoritmos para tomada de decisão dos agentes dependem disto para funcionar. Também não foi considerado o caso de redes cujo número de nodos varia — o que pode criar a necessidade de se ajustar a população de agentes dinamicamente. Por fim, a abordagem usada para o envelhecimento da informação só é aplicável em redes ad hoc, pois se baseia em uma previsão que leva em conta a mobilidade dos nodos e uma estimativa da distância física entre eles.

Capítulo 5

Uma Estratégia para a Descoberta da Topologia de Redes Dinâmicas e Descentralizadas com Agentes Móveis Inteligentes

Este trabalho propõe a utilização de agentes móveis inteligentes para a descoberta e manutenção, em cada nodo, de uma descrição da topologia da rede. Neste capítulo, essa proposta é detalhada. Na seção 5.1, é apresentado um modelo para redes dinâmicas como aquelas mostradas no capítulo 2. A seção 5.2 descreve o funcionamento da estratégia proposta de maneira geral. As seções 5.3 a 5.8 tratam de diversos aspectos desta estratégia.

5.1 Modelo da Rede

Redes dinâmicas e descentralizadas, como as redes P2P descentralizadas não estruturadas e redes móveis sem fio ad hoc, possuem uma série de características em comum. Pode-se destacar os seguintes atributos:

- *Descentralização*: não existe qualquer tipo de administração centralizada. Também não há um relógio global que seja compartilhado por todos os nodos.

- *Dinamismo*: tanto a configuração dos enlaces quanto os nodos presentes na rede variam de forma constante e imprevisível. Nodos e enlaces podem surgir, desaparecer e retornar à rede a qualquer momento. O tamanho da rede é indeterminado, podendo variar com o passar do tempo.
- *Arbitrariedade*: como consequência do dinamismo, nem sempre existe um caminho entre dois nodos quaisquer, ou seja, podem ocorrer particionamentos na rede e “sub-redes” isoladas podem ser formadas.
- *Heterogeneidade*: cada nodo da rede pode possuir propriedades (mobilidade, número de enlaces, largura de banda, memória, capacidade de processamento, etc.) diferentes dos demais.

O modelo de rede proposto por este trabalho assume redes que possuem essas características. A topologia da rede é então definida por um grafo $G(t) = (V_t, E_t)$ no qual, em um tempo t , cada vértice $v_{it} \in V_t$ representa um nodo i e cada aresta $e_{ijt} \in E_t$ representa um enlace entre um nodo i e um nodo j . Pode-se observar que a topologia da rede é uma função do tempo. Dados t_1 e t_2 tal que $t_1 \neq t_2$, $G(t_1)$ pode ser diferente de $G(t_2)$.

São feitas as seguintes suposições a respeito dos nodos da rede:

1. Um nodo é capaz de determinar de maneira inequívoca em um dado instante de tempo a sua vizinhança, ou seja, quais são os seus vizinhos. Para um nodo i , a vizinhança em um tempo t é definida como sendo $\{v_{jt} \in V_t | e_{ijt} \in E_t\}$.
2. Um nodo é capaz de enviar mensagens aos seus vizinhos sem erros em um tempo não desprezível, mas limitado. O tempo máximo, $1 < t_{msg} < \infty$, é conhecido. Se o enlace pelo qual uma mensagem está sendo enviada deixa de existir durante o envio, assume-se que a mesma retorna ao nodo de origem.
3. A escala de tempo é igual para todos os nodos, ou seja, apesar de os nodos não compartilharem um relógio global, a diferença entre um tempo t e o seu tempo anterior $t - 1$ é igual em todos os nodos.

4. Os enlaces entre nodos são bidirecionais, ou seja, para todo enlace $e_{ijt} \in E_t$ existe um correspondente $e_{jit} \in E_t$.

5.2 Funcionamento Básico

A estratégia proposta por este trabalho emprega agentes móveis inteligentes na descoberta da topologia da rede. Mais especificamente, a metáfora da colônia de formigas (descrita na seção 3.3) é seguida. Essa estratégia é em vários aspectos similar àquela apresentada em [20], descrita na seção 4.4. Esta seção descreve de maneira geral o funcionamento da estratégia proposta.

Os agentes que realizam a descoberta da topologia são chamados de agentes de descoberta. Eles migram entre os nodos disseminando informações, que são dadas na forma de grafos não direcionados que descrevem a topologia da rede, com vértices e arestas representando, respectivamente, nodos e enlaces. As informações que constam no grafo podem ser limitadas, para que o consumo de recursos pelos agentes seja controlado. Desta forma, a descrição da topologia carregada por cada agente pode ser parcial.

Cada nodo mantém localmente uma descrição da topologia da rede. Esta descrição é similar àquela carregada pelos agentes de descoberta, e sempre contém pelo menos o nodo, seus vizinhos, e os enlaces existentes entre eles. Outras informações devem ser trazidas ao nodo por agentes de descoberta. A descrição funciona de forma similar a um quadro negro (estratégia de comunicação apresentada na seção 3.2.2), permitindo a comunicação sematectônica entre agentes (categoria de estigmergia apresentada na seção 3.3.2). Ela é acessada por intermédio de um agente local, presente em cada nodo. Esse agente é responsável pelas interações entre agentes e nodos.

Os agentes de descoberta não possuem uma “origem” ou “destino” determinados, permanecendo na rede indefinidamente, até o seu desaparecimento ou a sua destruição por um agente local. A interação entre nodos e agentes é cíclica, com cada ciclo tendo os seguintes passos:

1. Um agente de descoberta chega a um nodo.

2. A descrição da topologia no nodo é atualizada pelo agente local com as informações trazidas pelo agente de descoberta.
3. O agente de descoberta permanece inativo por um período de tempo.
4. O agente de descoberta faz uma cópia da descrição da topologia.
5. O agente de descoberta migra para um dos vizinhos do nodo e o ciclo é reiniciado no nodo de destino.

Esta estratégia possui diversos aspectos que são discutidos nas seções seguintes.

5.3 Desatualização das Informações

Para que informações a respeito de um nodo e seus enlaces cheguem a nodos mais distantes que a sua vizinhança, é preciso que as mesmas sejam disseminadas pelos agentes de descoberta. Como a transmissão de mensagens entre nodos — e conseqüentemente a migração de agentes — ocorre em um tempo não desprezível, existe um intervalo de tempo entre o instante em que uma informação é gerada e o seu recebimento por um nodo. Como a rede é dinâmica, é possível que a sua topologia mude durante este intervalo, e um nodo pode receber informações que já não correspondem mais à realidade. Esse é o problema da desatualização das informações, ilustrado na figura 5.1.

O dinamismo da rede também pode desatualizar uma informação que já está presente na descrição da topologia em um nodo. Isto ocorre se a topologia sofre alterações sem que o nodo fique ciente das mesmas. Esta situação é ilustrada na figura 5.2.

Deve-se observar que um número arbitrário de alterações pode ocorrer na topologia da rede no intervalo de tempo em que um agente migra de um nodo para outro. Portanto, é possível afirmar que a rede sempre terá nodos cujas informações estarão sujeitas à desatualização. Quanto maior for o dinamismo da rede e o tempo necessário para uma informação ser levada de um nodo a outro, maior será a probabilidade de que a desatualização ocorra. Como não é possível para os nodos determinar com certeza a topologia da rede, este trabalho propõe a criação de mecanismos para a quantificação da confiabilidade

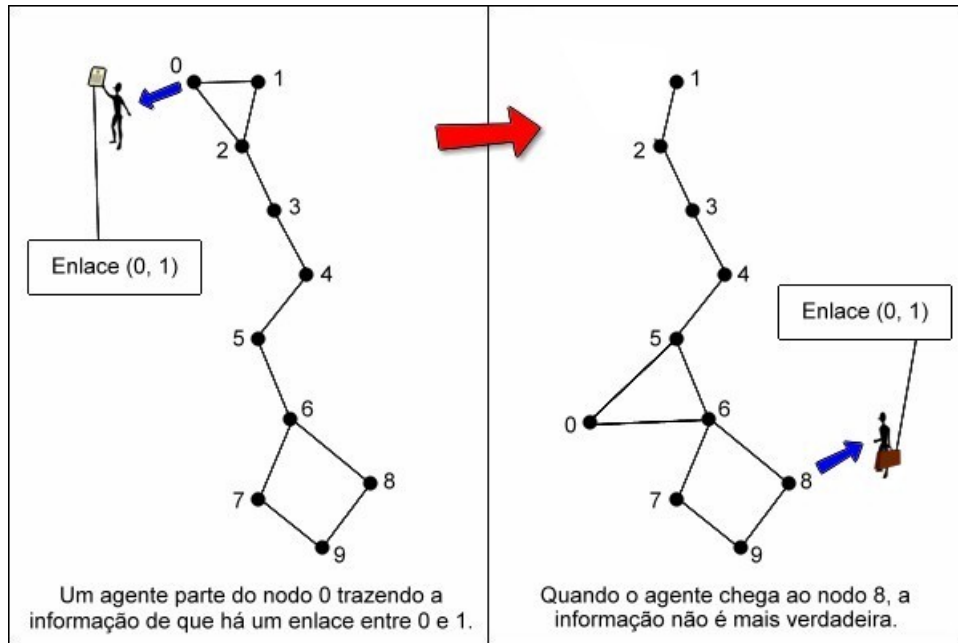


Figura 5.1: Um nodo recebe uma informação desatualizada.

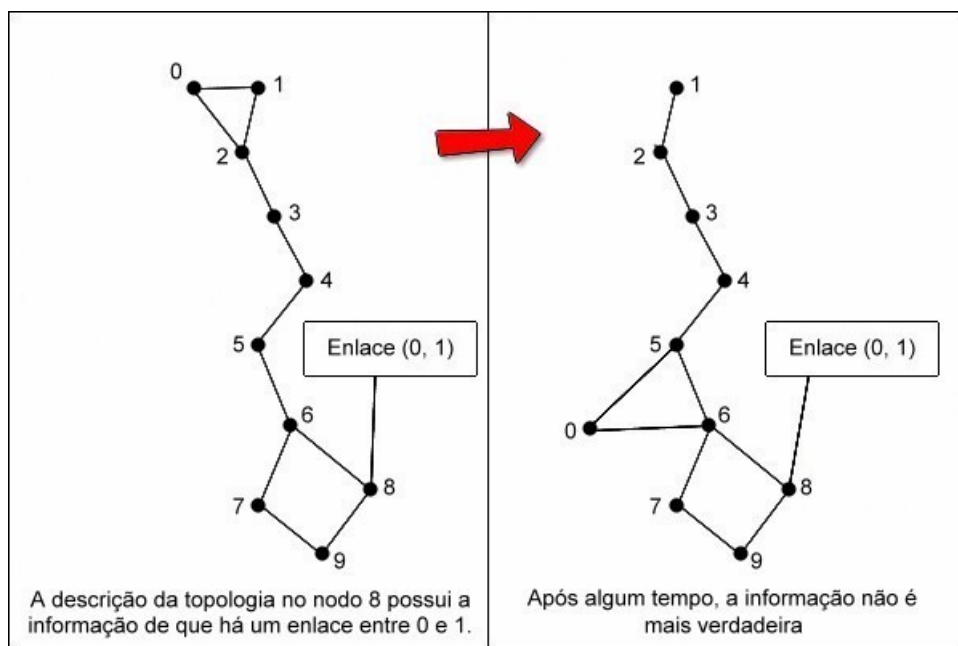


Figura 5.2: A descrição da topologia em um nodo se torna desatualizada.

de cada informação, com base em uma estimativa da sua idade. Estes mecanismos são discutidos a seguir.

5.4 Idade e Envelhecimento das Informações

Dado o problema da desatualização das informações, é preciso que a descrição da topologia em cada nodo contenha as informações mais recentes que puderem ser obtidas, pois estas informações possuem uma chance maior de estarem corretas. Desta forma, deve ser possível para nodos e agentes calcular ou estimar a “idade” de cada informação. A idade de uma informação é uma medida da sua confiabilidade: com passar do tempo, as informações “envelhecem”, tornando-se menos confiáveis.

A idade das informações é empregada de diversas formas. Uma descrição da topologia é atualizada apenas com as informações trazidas pelos agentes de descoberta que são mais recentes que aquelas já presentes na descrição. Além disso, informações cuja idade atinja um certo limite são removidas da descrição e descartadas pelos agentes de descoberta. Isso também permite que os nodos fiquem cientes de enlaces ou nodos que deixam de existir, já que as informações carregadas pelos agentes de descoberta limitam-se aos enlaces e nodos que existem na rede. O tempo máximo para que isso ocorra será igual à idade máxima que as informações da descrição da topologia podem ter. A idade das informações pode ser aproveitada até mesmo por outras aplicações que usam a descrição da topologia.

Como os nodos não compartilham um relógio global, e o tempo exato que um agente gasta para migrar entre dois nodos é desconhecido, não é possível determinar com exatidão a idade de uma informação. Por isso, ela deve ser estimada. Como dito na seção 5.1, o modelo de rede supõe que o tempo máximo necessário para que um agente migre entre dois nodos é conhecido. Este tempo máximo é acrescentado à idade das informações carregadas por um agente sempre que ele migra. Desta forma, a idade estimada para uma informação é sempre maior ou igual à idade real, e nenhuma informação é considerada mais recente do que realmente é. Já as informações presentes nas descrições da topologia nos nodos podem envelhecer de acordo com o relógio local do nodo, pois se supõe que a escala de tempo é igual em todos os nodos.

5.5 Migração

Dado o papel de disseminadores de informações dos agentes de descoberta, pode-se afirmar que o desempenho da estratégia proposta está diretamente ligado aos itinerários percorridos por estes agentes. Isto pode ser exemplificado através de uma situação extrema, ilustrada na figura 5.3. Nela, assume-se que todos os agentes que partem do nodo 1 migram para o nodo 2. Desta forma, as informações carregadas por esses agentes só poderão ser recebidas por outros nodos depois da sua passagem pelo nodo 2. O nodo 3, que é vizinho do nodo 1, só poderá receber essas informações depois que as mesmas tiverem passado por vários outros nodos. Mais que isto, os nodos 4, 5 e 6 jamais receberão essas informações, já que qualquer caminho entre eles e o nodo 2 passa pelo nodo 1.

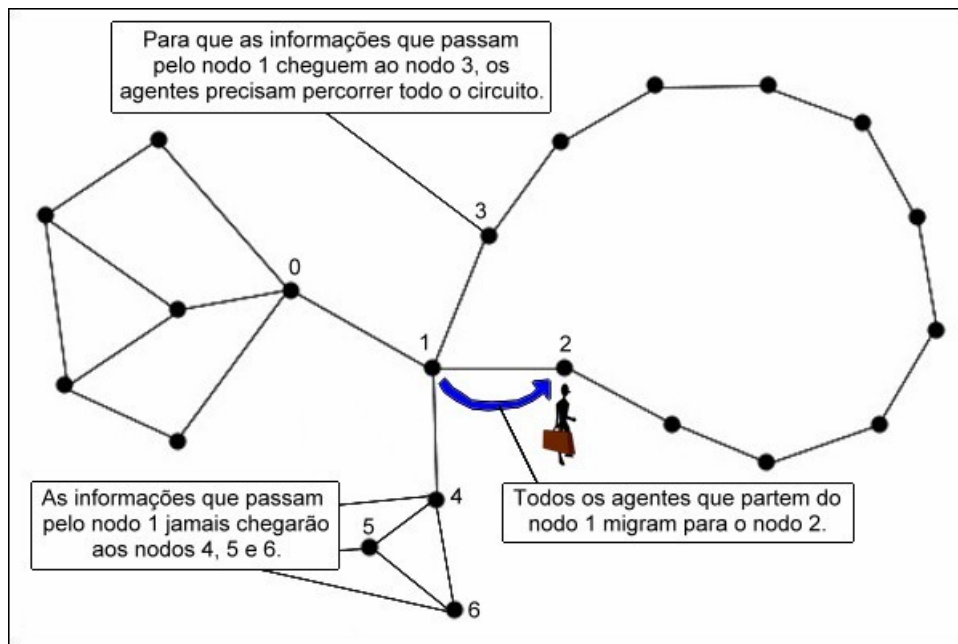


Figura 5.3: Um itinerário a ser evitado.

Outra observação pode ser feita a respeito da figura 5.3. Pode-se notar que todos os agentes que passarem pelo nodo 1 levarão ao nodo 2 as informações referentes à vizinhança do nodo 1. Estas informações só mudarão se a vizinhança do nodo 1 sofrer alguma alteração. Extrapolando esta idéia para mais nodos, pode-se induzir que, se os agentes seguirem todos um mesmo caminho, existe uma tendência de pouca ou nenhuma informação relevante ser levada aos nodos. Portanto, o objetivo dos agentes de descoberta deve

ser o de carregar as informações referentes a cada nodo para o maior número de nodos possível, ou seja, os agentes devem se “espalhar” pela rede.

A estratégia proposta por este trabalho assume que todas as decisões a respeito da migração dos agentes são tomadas pelos próprios agentes. Além disso, cada destino é escolhido no momento da migração, ou seja, o agente escolhe o próximo nodo do seu itinerário somente quando for migrar do nodo atual. Dado o modelo da rede, abordagens centralizadas ou nas quais todos os agentes fazem um planejamento em conjunto são inviáveis.

A maneira mais simples de fazer os agentes se espalharem pela rede é a seleção aleatória: o próximo destino de um agente é escolhido aleatoriamente entre os vizinhos do nodo atual. Este trabalho também propõe uma abordagem que usa a estigmergia baseada em sinais, descrita na seção 3.3.2. Os agentes se comunicam através de “feromônios” que são depositados nos nodos. De maneira similar à estratégia apresentada em [3] para o ADVR, estes feromônios possuem um funcionamento oposto ao tradicional, diminuindo a chance de um agente seguir o mesmo caminho que outro agente percorreu há pouco tempo, ou seja, os agentes “fogem” uns dos outros.

Cada nodo possui uma área para que, associados a cada um dos seus enlaces, feromônios sejam depositados. A área associada a um enlace é chamada de depósito de feromônio. Antes de migrar de um nodo, cada agente deixa no depósito associado ao nodo de destino uma unidade de feromônio. Assim, um agente que migre do nodo i para o nodo j atualiza a concentração relativa ao enlace e_{ijt} no depósito apropriado em i . Dada $C_{ij}(t)$ a concentração de feromônios relativa ao enlace e_{ijt} , a concentração é atualizada de acordo com a equação 5.1. Observa-se que a atualização é instantânea, ou seja, ocorre no mesmo instante em que o agente migra, t . Com isso, a migração de um agente afeta outros agentes que porventura venham a migrar simultaneamente.

$$C_{ij}(t) = C_{ij}(t) + 1 \quad (5.1)$$

Com o passar do tempo, os feromônios em um depósito “evaporam”, tornando-se cada vez mais fracos. A evaporação ocorre a uma taxa $0 \leq \rho \leq 1$ para cada unidade de tempo.

Quanto maior for o valor de ρ , mais rápida será a evaporação dos feromônios. Uma taxa de evaporação igual a 0 indica que os feromônios nunca enfraquecem, enquanto uma taxa igual a 1 indica que os feromônios afetam somente agentes que migram simultaneamente. A concentração em um tempo t é dada em função da concentração em um tempo anterior $t_0 < t$, de acordo com a equação 5.2.

$$C_{ij}(t) = C_{ij}(t_0) \cdot (1 - \rho)^{(t-t_0)} \quad (5.2)$$

Tendo definida a concentração de feromônios, deve-se também determinar a forma como eles afetam a migração dos agentes. A seleção do próximo destino de um agente é probabilística. A concentração de feromônios em cada depósito afeta a chance de escolha de cada enlace de acordo com uma constante $\alpha > 0$, que indica a “força” dos feromônios. Quanto maior for o valor de α , mais “fortes” eles serão, e maior será o peso dado a cada unidade de feromônio. Dados $|e_{it}|$ o número de vizinhos do nodo i , e $E_{it}(k)$ o k -ésimo vizinho do nodo i no tempo t , a probabilidade $P_{ij}(t)$ de escolha para o enlace e_{ijt} é dada pela equação 5.3.

$$P_{ij}(t) = \frac{(C_{ij}(t) + 1)^{-\alpha}}{\sum_{k=1}^{|e_{it}|} (C_{iE_{it}(k)}(t) + 1)^{-\alpha}} \quad (5.3)$$

5.6 População de Agentes de Descoberta

O modelo de rede considerado neste trabalho assume que não ocorrem erros durante a transmissão de mensagens entre nodos. Desta forma, agentes de descoberta não podem desaparecer da rede por causa de falhas ocorridas durante a migração. Se um enlace desaparece durante a migração de um agente através do mesmo, assume-se que o agente retorna ao nodo de origem, podendo escolher outro destino. Porém, ainda assim, agentes de descoberta podem desaparecer da rede. Isto ocorrerá se o nodo no qual um agente se encontra falhar ou sair permanentemente da rede. Além disso, o tamanho da rede pode variar, o que também muda o número de agentes que deve ser usado para a descoberta. Portanto, o tamanho da população de agentes de descoberta deve ser dinâmico e adaptável.

Este trabalho propõe que o processo de ajuste no tamanho da população de agentes se baseie na concentração dos mesmos feromônios que são usados para a migração. Quando a concentração atinge um limite mínimo, o nodo cria um novo agente. Para evitar que vários nodos criem agentes em um período de tempo muito curto, isso não é feito imediatamente, mas em um tempo aleatório dentro de um intervalo que vai de 0 a um tempo máximo. Da mesma forma, se a concentração atinge um limite máximo, todos os agentes que passam pelo nodo são destruídos antes de migrar.

A concentração que deve ser comparada aos limites mínimo e máximo, $C_i(t)$, é a concentração total de feromônios no nodo i , ou a soma das concentrações em todos os seus depósitos. Então, dados $|e_{it}|$ o número de vizinhos do nodo i ; $E_{it}(k)$ o k -ésimo vizinho do nodo i , e $C_{ij}(t)$ a concentração de feromônios associados a um enlace e_{ijt} no tempo t , a concentração total $C_i(t)$ é dada pela equação 5.4.

$$C_i(t) = \sum_{k=1}^{|e_{it}|} C_{iE_{it}(k)}(t) \quad (5.4)$$

Deve-se observar que uma quantidade maior de informações é gerada em nodos com muitos vizinhos. Estes nodos devem então criar mais agentes, e desta forma os limites mínimo e máximo são alterados pelo número de vizinhos de cada nodo de acordo com uma constante $\delta \geq 0$. Assim, dados um limite mínimo $\gamma_{min} > 0$ e um limite máximo $\gamma_{max} > \gamma_{min}$, a concentração total de feromônios $C_i(t)$ deve ser comparada aos valores $L_{min}(t)$ e $L_{max}(t)$, definidos respectivamente pelas equações 5.5 e 5.6.

$$L_{min}(t) = \gamma_{min} \cdot |e_{it}|^\delta \quad (5.5)$$

$$L_{max}(t) = \gamma_{max} \cdot |e_{it}|^\delta \quad (5.6)$$

Como o número de agentes criados por um nodo é influenciado pelo número de vizinhos do mesmo, observa-se que uma rede mais conectada (com um número maior de enlaces) deverá criar um número de agentes maior que uma rede menos conectada que tenha o mesmo número de nodos. Essa propriedade pode acabar ocasionando a criação de um

número excessivo de agentes em redes muito conectadas. Por outro lado, esse ajuste pelo número de vizinhos pode ser muito importante em redes nas quais apenas alguns poucos nodos têm um grande número de vizinhos.

Uma última observação pode ser feita a respeito desta abordagem para o ajuste da população de agentes. O instante de tempo no qual um nodo deverá começar a criar agentes pode ser previsto. Isso pode ser útil, por exemplo, para economizar recursos nos nodos, evitando que a concentração de feromônios tenha que ser verificada a todo instante. A criação de agentes deverá começar no instante t no qual a concentração $C_i(t)$, por conta da evaporação, atinge um valor igual a $L_{min}(t)$. Dado o tempo atual t_0 , t pode ser descoberto de acordo com a equação 5.7.

$$t = t_0 + \frac{\log(\frac{L_{min}(t_0)}{C_i(t_0)})}{\log(1 - \rho)} \quad (5.7)$$

Deve-se notar que esta previsão não pode ser aplicada a um nodo sem vizinhos. Neste caso, o nodo não deve criar agentes, já que os mesmos não teriam como disseminar informações. Além disso, um nodo cuja concentração de feromônios é igual a 0 deve criar agentes tão logo quanto possível. Outros casos especiais são aqueles nos quais a taxa de evaporação dos feromônios, ρ , é igual a 0 ou 1. No primeiro caso, os feromônios nunca evaporam, e agentes nunca são criados. Já no segundo caso, a concentração de feromônios chegará imediatamente a 0, fazendo com que o nodo crie um agente. Por fim, deve-se notar que a previsão só é válida se a concentração de feromônios não mudar por quaisquer outros meios que não a evaporação. Portanto, se o número de vizinhos do nodo muda, ou se um agente de descoberta migra do nodo, a previsão deve ser feita novamente.

5.7 Tempo Antes de Migrar

Durante o passo 3 do ciclo descrito na seção 5.2, o agente de descoberta permanece inativo no nodo por um período de tempo. Este período é chamado de “tempo antes de migrar”, ou TtM (*Time to Migrate*). O TtM é usado para controlar o tráfego de agentes na rede, reduzindo o consumo de recursos de cada agente individual. Se o tamanho da população de

agentes é mantido e o TtM é aumentado, o tráfego de agentes na rede diminui. Porém, esta redução no tráfego pode resultar em nodos com descrições que sofrem poucas atualizações, especialmente se a rede for muito dinâmica. Isso pode exigir um aumento no número de agentes de descoberta usados — o que pode de fato aumentar o tráfego de agentes na rede.

Ainda que isto ocorra, tal aumento pode se mostrar vantajoso, já que um número maior de agentes pode estar melhor distribuído entre os nodos, como mostra a figura 5.4. Nela, são apresentados dois cenários, c1 e c2, em dois instantes de tempo, t1 e t2. Em ambos os cenários, o tráfego de agentes entre t1 e t2 é igual, com a ocorrência de 5 migrações. Porém, em c1 existem 5 agentes de descoberta que migram uma vez cada, e em c2 um único agente migra 5 vezes. Deve-se observar que, embora o tráfego de agentes seja igual em ambos os casos, em c1 o consumo de recursos dos nodos é maior. Isto ocorre porque, mesmo inativos, os agentes consomem recursos.

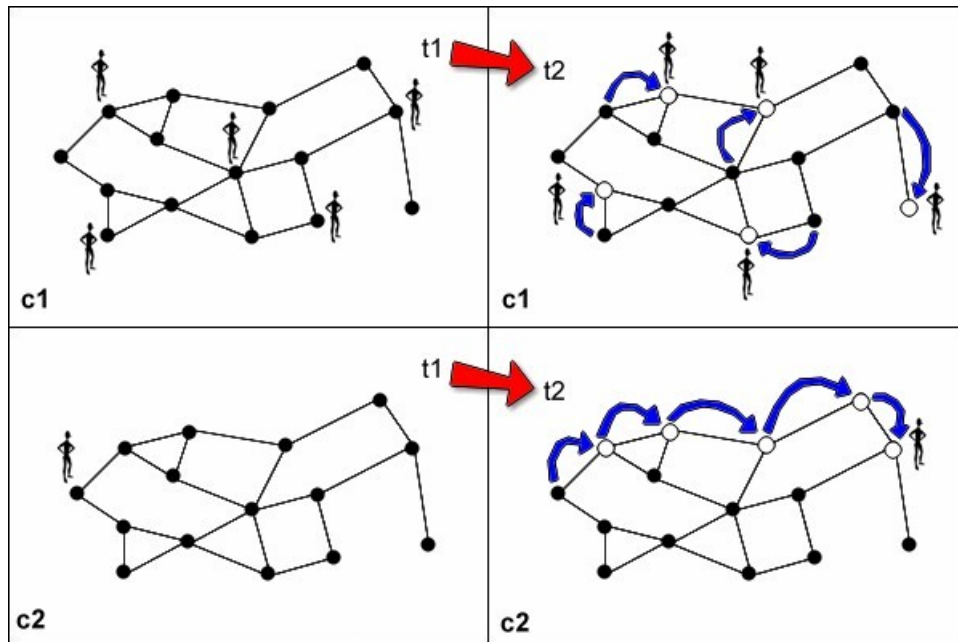


Figura 5.4: Distribuição dos agentes de descoberta com o TtM.

O TtM torna necessário o passo 4 do ciclo, no qual a descrição da topologia é copiada pelo agente de descoberta. Enquanto o agente está inativo, a vizinhança do nodo pode mudar, ou outros agentes de descoberta podem chegar ao nodo trazendo novas informações. Assim, a descrição da topologia pode ser alterada, e é preciso que o agente a copie

antes de migrar, para que a descrição mais recente seja levada a outros nodos. Além disso, o TtM deve ser acrescentado à idade das informações carregadas pelos agentes.

5.8 Alcance da Descrição

As características do modelo de rede considerado e da estratégia proposta têm um impacto direto sobre a descrição da topologia que pode ser obtida nos nodos. A desatualização das informações faz com que agentes e nodos precisem descartar informações cuja idade se torne muito alta. O tempo necessário para que um agente leve uma informação de um nodo a outro tende a aumentar de acordo com o número de migrações realizadas, por causa do tempo gasto na migração e do TtM. Portanto, há uma tendência de aumento na idade das informações conforme a distância entre os nodos aumenta. O número de nodos da rede é indeterminado, e é então possível que a distância entre dois nodos seja grande o suficiente para que todas as informações a respeito de um deles sejam descartadas antes de chegar ao outro.

Como exemplo, pode-se observar a situação mostrada na figura 5.5. Supõe-se que uma informação cuja idade chegue a 50 é descartada, e que a cada migração de um agente a idade das informações que o mesmo carrega é aumentada em 10. Os valores próximos aos nodos indicam a menor idade que uma informação gerada no nodo 0 pode ter ao chegar ao nodo. Pode-se observar que alguns nodos jamais receberão essa informação, pois sua idade atinge o valor máximo antes que a informação chegue a eles.

A distância entre os nodos também influi no tempo que uma informação pode ficar em cada descrição da topologia. Como a idade das informações tende a aumentar conforme aumenta a distância entre nodos, informações sobre nodos distantes tendem a ficar menos tempo na descrição da topologia em um nodo. Assim, a descrição da topologia em um nodo tende a conter mais informações a respeito de nodos mais próximos.

Pode-se observar que, em certos casos, não é possível manter nos nodos uma descrição completa da topologia. Nesses casos, cada nodo terá uma visão parcial da topologia. Como a tomada de decisões a respeito da migração dos agentes é feita de maneira probabilística, não é possível determinar de antemão o seu itinerário — e conseqüentemente a

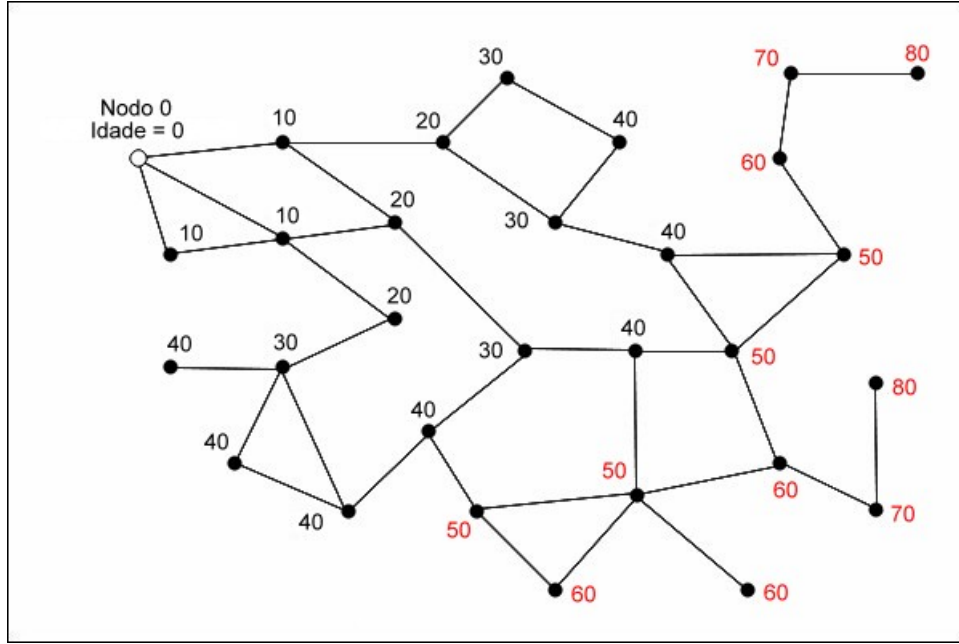


Figura 5.5: **Informações são descartadas antes de chegar a um nodo.**

distância que uma informação percorrerá antes de ser descartada. Portanto, não é possível determinar qual será o alcance exato da descrição obtida nos nodos. Descrições parciais da topologia são usadas, por exemplo, pelo YAPPERS [40], um sistema para a localização de recursos em redes P2P, no qual a área presente na descrição é chamada de vizinhança estendida de um nodo. Em [28] este tipo de descrição é chamado de mapa da área ao redor de um nodo.

A figura 5.6 mostra uma possível descrição da topologia da rede ilustrada na figura anterior. Esta é a descrição localizada no nodo 0. Pode-se observar que as informações a respeito de áreas próximas do nodo tendem a estar mais “definidas”, enquanto aquelas a respeito de áreas distantes estão mais “apagadas”, ou seja, são informações mais antigas — e maior probabilidade de estarem desatualizadas.

O alcance da descrição em cada nodo é influenciado pela idade máxima das informações que são aceitas antes de serem descartadas. Deve existir um equilíbrio entre esta idade máxima e o alcance. Se forem aceitas informações com idade muito alta, boa parte delas deverá estar desatualizada, e recursos da rede serão consumidos para disseminar informações que não correspondem à realidade. No extremo oposto, apenas as informações com a menor idade possível são aceitas. Neste caso, os nodos conheceriam apenas os seus

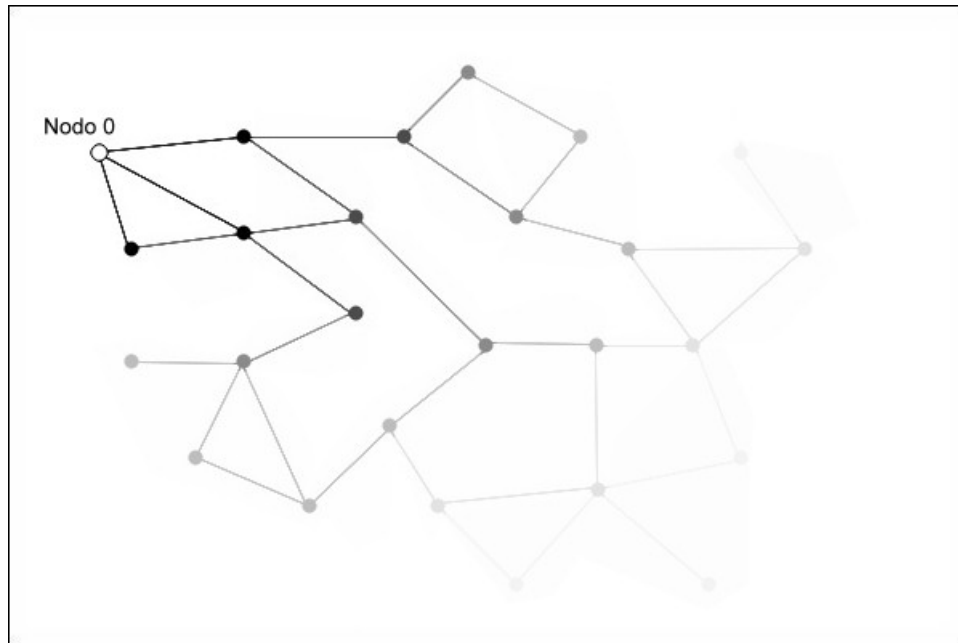


Figura 5.6: **Possível descrição da topologia da rede no nodo 0.**

próprios vizinhos, tendo uma visão extremamente restrita da rede. Por isso, valores como a idade máxima das informações e o TtM devem ser escolhidos com cautela.

Capítulo 6

Avaliação Experimental

A estratégia proposta no capítulo 5 para a descoberta da topologia foi implementada para que o seu desempenho pudesse ser avaliado. Ela foi testada em diferentes cenários em uma série de experimentos. Sobre os resultados destes experimentos, foi feita uma análise empírica do comportamento da estratégia em várias situações. Esta forma de avaliação foi escolhida por conta da grande dificuldade que existe em se avaliar analiticamente sistemas complexos, compostos por vários componentes autônomos que têm um comportamento estocástico e imprevisível [73].

Para que diferentes cenários pudessem ser considerados, a estratégia foi, em um primeiro momento, implementada em um ambiente simulado. A simulação também permite que a estratégia seja analisada isoladamente, sem que existam interferências externas. Além disso, esta abordagem torna possível a avaliação de situações que dificilmente poderiam ser criadas em uma implementação real, dada a sua complexidade ou custo. No entanto, deve-se ressaltar que várias das nuances e possibilidades de uma situação real são perdidas quando se transfere o ambiente para uma simulação. Por isso, apesar de neste trabalho a avaliação ter se limitado à simulação, uma implementação real se faz necessária futuramente.

Este capítulo descreve o ANTD, *software* desenvolvido para a simulação da estratégia proposta, e os experimentos realizados. Os resultados desses experimentos são então analisados e discutidos. A seção 6.1 apresenta o ANTD. São descritos o algoritmo para

a geração da topologia inicial (seção 6.1.1), os comandos que definem o comportamento dinâmico da rede (seção 6.1.2), os parâmetros que podem ser configurados (seção 6.1.3), e as métricas produzidas (seção 6.1.4). Já a seção 6.2, descreve os experimentos e seus resultados. Diversas características são avaliadas nas seções 6.2.1 a 6.2.6. Na seção 6.2.7, é feita uma síntese das conclusões dos resultados dos experimentos.

6.1 ANTĐ (*Agents for Network Topology Discovery*)

O ANTĐ (*Agents for Network Topology Discovery*) é o *software* desenvolvido para a avaliação da estratégia proposta por este trabalho para a descoberta da topologia. Trata-se de um simulador de agentes móveis em redes de topologia dinâmica e descentralizada, escrito na linguagem Java. A simulação é baseada em eventos, e é controlada por um agendador simples, que permite que eventos tenham sua ocorrência programada para instantes de tempo específicos, dados em uma escala de milissegundos. A ordem de chamada de eventos que ocorrem em um mesmo instante de tempo é arbitrária.

A arquitetura apresentada na seção 3.2.3 foi implementada nos nodos de uma maneira bastante simples. As características do ambiente de execução são simuladas através do agendamento de eventos por parte do ambiente computacional dos nodos. A interface de transporte oferece rotinas para que um agente “migre” entre nodos; migração esta que consiste na transferência do controle sobre o agente de um nodo para outro. A interface de comunicação oferece ao agente acesso indireto à descrição da topologia e aos depósitos de feromônios associados aos vizinhos do nodo. Dado o caráter experimental do ANTĐ, não são implementados quaisquer mecanismos de segurança ou interfaces para o acesso a recursos do nodo. Da mesma forma, os agentes locais não foram implementados como tal, tendo sua funcionalidade distribuída entre os componentes do ambiente computacional.

As próximas seções detalham várias das características do ANTĐ. A topologia inicial das redes simuladas é gerada aleatoriamente por um algoritmo descrito na seção 6.1.1. As transformações sofridas por essas topologias são definidas por arquivos contendo eventos, descritos na seção 6.1.2. Os parâmetros da simulação da estratégia que podem ser configurados são apresentados na seção 6.1.3. Por fim, na seção 6.1.4 são definidas as métricas

que o programa usa para a avaliação da estratégia.

6.1.1 Topologia Inicial

Para que as simulações feitas com o ANTD sejam significativas, é preciso que as redes simuladas tenham topologias com características similares às aquelas encontradas em redes reais. Estudos realizados sobre a topologia da Internet identificaram diversos aspectos que seguem distribuições do tipo *power law* [31], ou algum outro tipo de distribuição de cauda pesada [18]. Isso pode ser observado tanto na rede formada pelos roteadores quanto naquela formada pelos domínios. Entre as características que seguem distribuições de cauda pesada, pode-se destacar o número de vizinhos de cada nodo, ou seja, existem alguns poucos nodos que possuem um grande número de vizinhos, enquanto a maior parte dos nodos possui apenas um ou dois. O mesmo tipo de distribuição pode ser observado na rede *peer-to-peer* Gnutella [88]. Por esta razão, as topologias geradas pelo ANTD possuem aspectos que seguem distribuições de cauda pesada.

A geração da topologia de uma rede simulada com propriedades que seguem distribuições de cauda pesada é um problema conhecido, para o qual diversas soluções foram propostas [95, 71]. O ANTD usa o algoritmo proposto por Bu e Towsley em [12] para gerar a topologia inicial da rede. Esse algoritmo recebe como parâmetros:

- n : O número de nodos da rede.
- m_0 : O número de nodos do *backbone* inicial. Deve ser menor que n .
- $M()$: Uma função que retorna, de acordo com certa distribuição, o número de enlaces criados a cada passo do algoritmo. O valor máximo deve ser menor ou igual a m_0 . Em [12], esta função tem uma chance de 87% de retornar 1 e 13% de retornar 2. O ANTD mantém essa mesma função.
- p : A probabilidade de um nodo ser criado a cada passo do algoritmo. O valor de p deve ser maior que 0 e menor ou igual a 1.

- β : Parâmetro que indica a preferência dada a nodos com muitos vizinhos quando um enlace é criado. Quanto menor for o valor de β , menor será a preferência. O valor de β deve ser inferior a 1.

Dados esses parâmetros, o algoritmo é o seguinte:

1. A rede é iniciada com um *backbone* de m_0 nodos conectados por $m_0 - 1$ enlaces.
2. Com probabilidade p , um novo nodo é acrescentado à rede com $M()$ enlaces o conectando a nodos já existentes; ou com probabilidade $1 - p$, $M()$ novos enlaces são criados entre nodos já existentes. Quanto menor for o valor de p , mais conectada deverá ser a rede, pois o número de enlaces criados será maior. Em ambos os casos, os nodos aos quais os novos enlaces são conectados são escolhidos de acordo com a função Π , dada pela equação 6.1, descrita abaixo.
3. Repete o passo 2 até que o número de nodos na rede seja igual a n .
4. Remove da rede quaisquer enlaces repetidos ou que conectem um nodo a si próprio.

A escolha dos nodos aos quais os enlaces se conectam é feita de acordo com Π , uma função de preferência linear generalizada (Generalized Linear Preference — GLP). Esta função determina a chance de um nodo i da rede ser escolhido. Dado $|e_i|$ igual ao número de vizinhos de um nodo i , $\Pi(i)$ é dado pela equação 6.1.

$$\Pi(i) = \frac{|e_i| - \beta}{\sum_{j=0}^n (|e_j| - \beta)} \quad (6.1)$$

Os parâmetros do algoritmo de Bu e Towsley podem ser configurados no ANTD na tela mostrada na figura 6.1.

Pode-se observar que, além dos parâmetros para o algoritmo, são apresentados na região inferior da tela parâmetros para a geração de arquivos de eventos. Os arquivos de eventos e o gerador são descritos a seguir.

Figura 6.1: Tela do ANTD para a configuração de parâmetros da rede.

6.1.2 Dinamismo

O comportamento dinâmico das redes simuladas no ANTD é definido por arquivos de entrada, que contêm séries de comandos para o agendamento de eventos. Estes comandos, descritos abaixo, permitem a criação e destruição de nodos e enlaces.

- `addLink (time, node1, node2)`: Cria um enlace entre dois nodos.
- `removeLink (time, node1, node2)`: Remove o enlace entre dois nodos.
- `addNode (time, node)`: Cria um nodo com o identificador dado.
- `removeNode (time, node)`: Remove da rede o nodo com o identificador dado e todos os enlaces que o conectam aos seus vizinhos.
- `addNodeAndLink (time, node, neighbor)`: Cria um nodo com o identificador dado e o conecta a um vizinho.

Caso um comando não possa ser executado (por exemplo, se um comando tenta criar um nodo que já existe), a topologia da rede permanece inalterada. Deve-se observar

também que esse conjunto de comandos torna possível a geração e o controle de topologias específicas. Para isso, pode-se gerar a rede inicial com um número mínimo de nodos, agendando a criação dos demais nodos e enlaces para o início da simulação.

Além do instante de tempo no qual os eventos ocorrem, os comandos recebem como parâmetros identificadores de nodos. Como a rede é gerada automaticamente, a identificação de um nodo ou enlace específico pode ser impossibilitada. A rede inicial possui os nodos 0 a $n - 1$, mas não se pode inferir mais a respeito das características desses nodos. Por isso, os comandos podem receber como parâmetro um dos seguintes “curingas”:

- **AVAILABLE:** Um nodo inexistente, com o menor identificador que não esteja sendo usado. Deve ser usado apenas para simular o retorno de um nodo à rede.
- **NEW:** Um nodo inexistente, com o menor identificador que ainda não tenha sido usado. Deve ser usado apenas na criação de novos nodos.
- **HRANK:** O nodo com o maior número de vizinhos.
- **LRANK:** O nodo com o menor número de vizinhos.
- **RANDOM:** Um nodo arbitrário, escolhido aleatoriamente.
- **NEIGHBOR:** Um nodo arbitrário, vizinho de outro nodo. Deve ser usado exclusivamente na remoção de enlaces.
- **GLP:** Um nodo arbitrário, escolhido de acordo com a função GLP (eq. 6.1).
- **HPOP:** O nodo com o maior número de agentes.
- **LPOP:** O nodo com o menor número de agentes.

Todos os curingas se referem à rede no momento em que o evento ocorre. Caso existam dois ou mais nodos que se enquadrem em uma categoria, um dos nodos é escolhido arbitrariamente.

Na figura 6.1, mostrada na seção anterior, pode-se observar a existência de uma região com parâmetros para a geração automática de arquivos de eventos. Este gerador cria automaticamente arquivos contendo comandos para a criação e remoção de nodos e enlaces.

O valor configurável no ANTD é o tempo máximo entre a ocorrência de dois comandos de um mesmo tipo. Desta forma, o valor 240000 para o comando `addNodeAndLink` indica que um novo nodo é criado e conectado à rede em intervalos de, no máximo, 240000 ms, ou 4 minutos. O tempo exato é definido aleatoriamente, seguindo uma distribuição uniforme; e os comandos são gerados até que o tempo atinja um limite dado. Os outros parâmetros que os comandos recebem são sempre os mesmos. Eles foram escolhidos de forma a simular um comportamento dinâmico sem que a topologia perca as suas características que seguem distribuições do tipo *power-law*.

6.1.3 Configuração da Estratégia para a Descoberta da Topologia

A estratégia para a descoberta da topologia possui diversos parâmetros que podem ser configurados. No ANTD, a configuração desses parâmetros é feita através da tela mostrada na figura 6.2.

Figura 6.2: Tela do ANTD para a configuração de parâmetros da estratégia para a descoberta da topologia.

Esses parâmetros são listados e descritos abaixo. Todos os tempos são dados em milissegundos.

- **Idade Máxima das Informações Carregadas Pelos Agentes:** Os agentes levam de um nodo para outro apenas informações cuja idade seja menor ou igual a este valor. Parâmetro descrito na seção 5.4.
- **Idade Máxima das Informações nos Nodos:** Informações cuja idade atinja este limite são descartadas da descrição da topologia na qual elas estiverem. Parâmetro descrito na seção 5.4.
- **Tempo Máximo para Migração:** Este é o tempo máximo que um agente leva para migrar entre dois nodos. O valor exato é sorteado de acordo com uma distribuição uniforme, estando entre 1 e o valor deste parâmetro. Parâmetro definido no modelo da rede, na seção 5.1.
- **Tempo Antes de Migrar:** Este é o tempo que um agente permanece inativo antes de migrar de um nodo. Parâmetro discutido na seção 5.7.
- **Método para Seleção de Destino:** Indica o método usado pelos agentes para selecionar o próximo nodo a ser visitado. Pode ser aleatório ou baseado em feromônios. Estes métodos foram discutidos na seção 5.5.
- **Força dos Feromônios:** Este é o parâmetro α , que influencia a escolha dos agentes quando a seleção de destino baseada em feromônios é usada, como dito na seção 5.5. Quanto maior for o seu valor, maior é a preferência dada a enlaces com baixa concentração de feromônios.
- **Taxa de Evaporação dos Feromônios:** Este é o parâmetro ρ , que define o quanto a concentração de feromônios associados a cada enlace é reduzida a cada unidade de tempo. Deve-se observar que esta taxa de evaporação assume uma escala de tempo de 100 ms. Parâmetro definido na seção 5.5.
- **População Inicial de Agentes de Descoberta:** Diz o número de agentes de descoberta que existem na rede no início da simulação. Deve ser maior que 0 se o ajuste dinâmico na população de agentes não estiver sendo usado. A população de agentes de descoberta foi discutida na seção 5.6.

- **Posicionamento da População Inicial de Agentes de Descoberta:** Diz como a população inicial de agentes de descoberta se distribui na rede. Os agentes podem estar igualmente distribuídos entre os nodos, ou os nodos podem ser escolhidos de acordo com a função GLP, definida na seção 6.1.1. No segundo caso, nodos com mais vizinhos tendem a ter um número maior de agentes no início da simulação.
- **Usar População Dinâmica de Agentes:** Diz se a estratégia para o ajuste automático da população de agentes, descrita na seção 5.6, será usada durante a simulação.
- **Limite Inferior para a Concentração de Feromônios:** Este é o parâmetro γ_{min} , que define a menor concentração de feromônios que um nodo pode ter antes de começar a criar agentes. Parâmetro descrito na seção 5.6.
- **Limite Superior para a Concentração de Feromônios:** Este é o parâmetro γ_{max} , que define a maior concentração de feromônios que um nodo pode ter antes de começar a destruir agentes. Parâmetro descrito na seção 5.6.
- **Peso da Vizinhança:** Este é o parâmetro δ , que altera de acordo com o número de vizinhos de cada nodo os limites inferior e superior para a concentração de feromônios. Parâmetro definido na seção 5.6.
- **Intervalo Máximo para a Criação de Agentes:** Quando a concentração de feromônios em um nodo chega ao limite mínimo, a criação de agentes não é iniciada imediatamente. O próximo agente é criado em um intervalo que vai de 0 a este tempo máximo. Parâmetro descrito na seção 5.6.

6.1.4 Métricas

Tendo definidos os parâmetros para a geração da topologia inicial e para a descoberta em si, o ANTD pode executar as suas simulações. Estas simulações são iniciadas a partir da tela mostrada na figura 6.3.

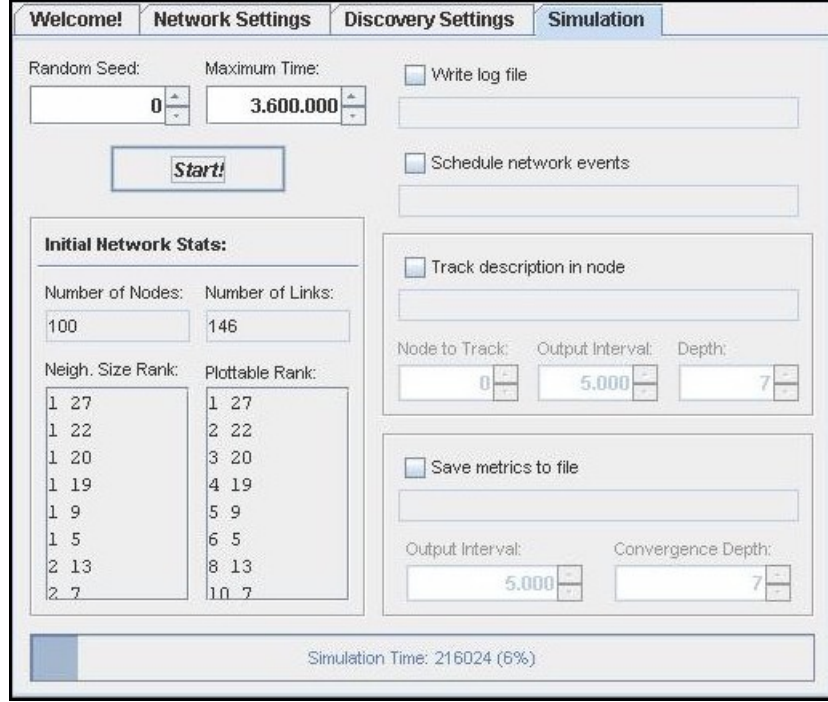


Figura 6.3: Tela do ANTD para a execução de simulações.

As simulações do ANTD produzem resultados numéricos que podem ser usados para avaliar o desempenho da estratégia proposta. As duas métricas mais importantes quantificam a diferença entre as descrições da topologia nos nodos e a topologia real da rede. Essas métricas, a convergência e a divergência da descrição, são descritas abaixo.

A convergência de uma descrição da topologia mede a proporção de enlaces que existem na rede em um dado momento e que estão presentes na descrição. Esta métrica é baseada na “convergência média da conectividade”, apresentada em [20]. A convergência é dada pelo número de enlaces que existem ao mesmo tempo na topologia e na descrição, dividido pelo número total de enlaces da rede. Por exemplo, se a rede possui 100 enlaces e 80 deles aparecem em uma descrição, a convergência é igual a 0.8, ou 80%. Dados E_t o conjunto de enlaces da topologia da rede no tempo t , e D_{it} o conjunto de enlaces da descrição no nodo i , a convergência é definida como:

$$Convergência(i, t) = \frac{|\{e_{ijt} | e_{ijt} \in E_t, e_{ijt} \in D_{it}\}|}{|E_t|} \quad (6.2)$$

Algumas observações devem ser feitas sobre o cálculo da convergência de uma des-

crição. Se a rede é particionada, de forma que existam nodos entre os quais não há um caminho, apenas a subrede na qual o nodo está é considerada no cálculo da convergência. Isso é feito desta forma porque, em uma rede desconexa, é impossível de se obter informações a respeito de certos enlaces. Neste caso, a falta da informação na descrição não está relacionada à estratégia usada para a descoberta, e sim a uma condição inerente à configuração atual da rede. A convergência também considera apenas os enlaces que estão até uma certa distância do nodo no qual a descrição está localizada. Esta distância máxima pode ser usada para avaliar o alcance das descrições que são obtidas nos nodos.

A divergência de uma descrição da topologia mede o oposto da convergência: a proporção de enlaces que existem na descrição mas que não existem na topologia real da rede. Esses “erros” surgem quando a topologia da rede muda e a descrição se torna desatualizada. A divergência é dada pelo número de enlaces que existem na descrição mas não existem na rede, dividido pelo número total de enlaces na descrição. Assim, se a descrição possui 100 enlaces e apenas 80 deles existem na rede de fato, a divergência é igual a 0.2, ou 20%. A divergência da descrição da topologia no nodo i no tempo t é definida como:

$$Divergência(i, t) = \frac{|\{e_{ijt} | e_{ijt} \in D_{it}, e_{ijt} \notin E_t\}|}{|D_{it}|} \quad (6.3)$$

Deve-se observar que essas métricas são complementares, e uma avaliação da estratégia para a descoberta da topologia deve levar em conta ambas. Isto pode ser ilustrado por dois casos extremos. No primeiro caso, a descrição tem todos os enlaces possíveis, ou seja, assume-se uma rede totalmente conectada. Neste caso, a convergência será sempre 1, pois todos os enlaces que existem na rede estarão certamente na descrição. Porém, a divergência será alta, já que muitos dos enlaces que estão na descrição não existirão na rede — a não ser que a rede seja de fato completamente conectada. No outro extremo, a descrição da topologia é vazia. Neste caso, ambas as métricas terão valor igual a 0.

Portanto, o objetivo da estratégia para a descoberta da topologia deve ser produzir nos nodos descrições com uma convergência alta e uma divergência baixa. Em uma situação ótima, todos os nodos possuem descrições com convergência igual a 1 — indicando a presença de todos os nodos da rede — e divergência igual a 0 — indicando que não há

informações errôneas.

O ANTD produz arquivos de saída contendo, entre outras métricas, a convergência e a divergência média das descrições da topologia. Estas medidas são feitas em intervalos de tempo regulares, e fazem a média considerando todos os nodos da rede. Outras métricas apresentadas nos arquivos de saída são o número total de agentes, nodos e enlaces presentes na rede no instante da medição.

Outras métricas produzidas pelo ANTD são: o número total de migrações de agentes que ocorreram durante a simulação, e o número médio de migrações que ocorreram a cada segundo. Este segundo valor pode ser usado para calcular o número médio de migrações de agentes que ocorreram em cada nodo a cada segundo. Para isso, basta dividir o número médio de migrações por segundo pelo número médio de nodos que a rede teve durante a simulação. Além dessas métricas, o programa permite o acompanhamento da descrição da topologia em um nodo específico, assim como a sua convergência e divergência com o passar do tempo.

6.2 Experimentos e Resultados

Vários experimentos foram realizados com o ANTD, com diferentes conjuntos de parâmetros, de forma que a estratégia proposta por este trabalho pudesse ter o seu desempenho avaliado em um grande número de situações. Em sua maioria, os resultados apresentados são uma média dos resultados de uma série de execuções. Além disso, algumas das métricas consideram a média dos resultados de todos os nodos da rede, enquanto outras consideram a média obtida no decorrer do tempo. Por isso, a análise dos resultados dos experimentos é empírica, e baseada nas tendências centrais dos resultados produzidos.

Os experimentos foram realizados, em sua maioria, com o intuito de analisar as consequências da variação dos valores de certos parâmetros. Quando o valor de um parâmetro muda, os demais parâmetros permanecem fixos. Assim, é definido um conjunto de “valores padrão” para os parâmetros. Caso um experimento não mencione o valor de um parâmetro, assume-se que o valor padrão é usado. Os valores padrão para os parâmetros são os seguintes:

- Idade Máxima das Informações Carregadas Pelos Agentes: 1 minuto.
- Idade Máxima das Informações nos Nós: 1 minuto.
- Tempo Máximo para Migração: 500 milissegundos.
- Tempo Antes de Migrar: 200 milissegundos.
- Método para Seleção de Destino: seleção baseada em feromônios.
- Força dos Feromônios (α): 5.
- Taxa de Evaporação dos Feromônios (ρ): 0.01.
- População Inicial de Agentes de Descoberta: 0 agentes.
- Posicionamento da População Inicial de Agentes de Descoberta: aleatório.
- Usar População Dinâmica de Agentes: verdadeiro.
- Limite Inferior para a Concentração de Feromônios (γ_{min}): 0.1.
- Limite Superior para a Concentração de Feromônios (γ_{max}): 3.4.
- Peso da Vizinhança (δ): 1.6.
- Intervalo Máximo para a Criação de Agentes: 1 segundo.

Caso não haja afirmação em contrário, a convergência é medida considerando a rede com uma distância de até 7 enlaces do nó no qual está a descrição avaliada. As métricas são normalmente produzidas em intervalos de 5 segundos.

Vários experimentos foram realizados sobre dois conjuntos de “redes padrão”. O primeiro conjunto tem 10 redes com $n = 100$ e $m_0 = 5$, e o segundo tem 10 redes com $n = 200$ e $m_0 = 10$. Ambos os conjuntos consideram os parâmetros abaixo, com cada valor de p e β sendo usado para uma das 10 redes do conjunto.

1. $p = 1.0$ e $\beta = 0$;
2. $p = 0.8$ e $\beta = 0$;

3. $p = 0.6$ e $\beta = 0$;
4. $p = 1.0$ e $\beta = 0.2$;
5. $p = 0.8$ e $\beta = 0.2$;
6. $p = 0.6$ e $\beta = 0.2$;
7. $p = 1.0$ e $\beta = 0.4$;
8. $p = 0.8$ e $\beta = 0.4$;
9. $p = 0.6$ e $\beta = 0.4$;
10. $p = 0.4$ e $\beta = 0.6$;

O gerador de arquivos de eventos do ANTD foi então usado para definir o comportamento dinâmico de cada uma das redes. Para as redes do primeiro conjunto, o tempo máximo para a criação de um nodo é de 5 minutos, para a remoção de um nodo é de 10 minutos, para a criação de um enlace é de 3 minutos, e para a remoção de um enlace é de 6 minutos. Os tempos para as redes do segundo conjunto são: 4 minutos para a criação de um nodo, 8 minutos para a remoção, 2 minutos para a criação de um enlace e 4 minutos para a remoção. Pode-se observar que, em ambos os casos, existe uma tendência de crescimento da rede inicial.

Todas as simulações realizadas sobre as redes padrão usam também um mesmo conjunto de sementes para o gerador de números aleatórios. Desta forma, obtém-se uma mesma topologia inicial, e os eventos ocorrem nos mesmos instantes de tempo. Porém, como os comandos se baseiam em “curingas”, os nodos e enlaces afetados podem mudar a cada simulação. De qualquer forma, como o objetivo dos experimentos é detectar tendências centrais, estas variações não devem ter um grande impacto na análise dos resultados como um todo, sendo até mesmo desejáveis em alguns casos.

As seções 6.2.1 a 6.2.6 discutem os experimentos que foram realizados, divididos em várias categorias. A última seção (6.2.7) traz um resumo dos resultados obtidos nos experimentos como um todo.

6.2.1 Qualidade das Descrições da Topologia

Uma primeira avaliação da estratégia proposta deve observar a qualidade das descrições da topologia que são obtidas nos nodos. Para isso, são definidos dois experimentos. No primeiro deles, são usadas as 10 redes do primeiro conjunto de redes padrão, que têm 100 nodos na topologia inicial. No segundo experimento, são usadas as redes do segundo conjunto, que têm 200 nodos na topologia inicial. Cada rede foi usada em uma simulação com duração de 1 hora. Os valores padrão foram usados para todos os parâmetros.

Antes dos resultados dos experimentos poderem ser avaliados, é preciso determinar quais medidas de tendências centrais serão usadas. Deve-se observar que no início das simulações não existem agentes de descoberta na rede, e as descrições da topologia nos nodos não contêm quaisquer informações. Por isso, todas as medidas de tendências centrais devem expurgar os resultados obtidos antes de um certo tempo, no qual se considera que o sistema está em um estado “instável”. Nestes experimentos, foram expurgados os resultados obtidos em tempos inferiores a 10 segundos.

Também é importante conhecer o número de observações disponíveis para cada métrica. No caso de resultados que são produzidos em intervalos regulares de tempo, como a convergência e a divergência média das descrições, cada simulação produz um resultado para cada intervalo de tempo. Como cada experimento se baseou em 10 execuções, se o tempo da medição for considerado, cada métrica possui 10 resultados para cada intervalo de tempo. Se o tempo da medição não for considerado, devem-se observar todos os resultados para uma determinada métrica. Como as simulações tiveram duração total de 1 hora, ou 3600 segundos, e as medições foram feitas em intervalos de 5 segundos, cada execução produziu um total de 720 resultados para cada métrica. Expurgando-se o primeiro resultado, que é obtido antes dos 10 segundos, tem-se 719 resultados em 10 simulações, ou seja, 7190 resultados para cada métrica.

Isso posto, foi feita uma análise da variação dos resultados dos experimentos considerando o instante da medição. Para isso, calculou-se o coeficiente de variação (divisão do desvio padrão pela média) das métricas que são produzidas em intervalos de tempo. Cada coeficiente de variação foi calculado com base nos resultados obtidos para cada métrica

em cada intervalo de tempo. Por exemplo, para se determinar o coeficiente de variação da convergência do experimento 1 aos 30 segundos, são usadas as convergências de cada uma das 10 simulações no tempo 30 segundos. O maior coeficiente de variação de cada uma das métricas foi então encontrado. Os coeficientes máximos são mostrados na tabela 6.1.

	Convergência	Divergência	No. de Agentes	No. de Nodos	No. de Enlaces
Experimento 1	2.4155%	316.2278%	63.8044%	3.2782%	19.0324%
Experimento 2	2.6453%	316.2278%	53.5535%	1.7753%	20.9909%

Tabela 6.1: Coeficientes de variação máximos, considerando o tempo da medição.

Esses resultados indicam que duas das métricas apresentam pouca variação, podendo ter o seu comportamento no tempo aproximado pela média dos resultados nas 10 simulações: a convergência e o número de nodos. O número de enlaces apresentou uma variação maior, mas ainda assim pode ser aproximado pela média. Já o número de agentes apresentou uma variação razoavelmente alta, e por este motivo, o seu comportamento no tempo será ilustrado pela mediana.

A divergência apresentou uma variação significativa. Esta grande variação ocorre porque a divergência normalmente tem um valor próximo de 0, o que distorce o coeficiente de variação. Por isso, o coeficiente de variação máximo da divergência foi recalculado considerando o valor 1 subtraído da divergência média para cada intervalo de tempo. Os coeficientes máximos foram 0.3951% para o experimento 1 e 0.2084% para o experimento 2 — indicando uma variação bastante baixa. Portanto, a divergência no tempo também pode ser aproximada pela média das 10 simulações.

Outra forma de se avaliar a estratégia é através de medidas de tendências centrais que valem para todos os instantes de tempo. Por isso, o coeficiente de variação das métricas também foi calculado considerando-se a média de todos os resultados, independente do tempo. Desta forma, o coeficiente de variação de cada métrica se baseia em uma média de 7190 valores. Os coeficientes de variação calculados são mostrados na tabela 6.2.1. Deve-se observar que, neste caso, a divergência média já está sendo subtraída de 1.

Pode-se notar que a variação não é especialmente alta para qualquer uma das métricas.

	Convergência	1 - Divergência	No. de Agentes	No. de Nós	No. de Enlaces
Experimento 1	0.6496%	1.2042%	26.3937%	4.0494%	13.454%
Experimento 2	0.5674%	0.6909%	20.7738%	2.3551%	18.0543%

Tabela 6.2: Coeficientes de variação, independente do tempo da medição.

Por isso, as medidas de tendências centrais usadas para os resultados no tempo serão repetidas: a mediana para o número de agentes e a média para as demais métricas.

Além das métricas calculadas em intervalos de tempo, o ANTD produz outros valores que podem ser usados para avaliar o desempenho da estratégia proposta. Estes valores são o número total de migrações de agentes e o número médio de migrações por segundo. Como o primeiro valor é dependente do tempo total da simulação, optou-se por usar o segundo valor. O coeficiente de variação do número médio de migrações por segundo é de aproximadamente 14% para o experimento 1 e 20% para o experimento 2, o que permite o uso da média como medida de tendência central.

Tendo definidas as medidas de tendência central que serão usadas, podem-se avaliar os resultados dos experimentos como um todo. Para isso, devem ser analisadas a convergência e a divergência das descrições produzidas pela estratégia. O gráfico da figura 6.4 mostra a convergência média obtida no decorrer do tempo para os dois experimentos.

A convergência média foi de 0.9953 para o experimento 1 e de 0.9957 para o experimento 2. Junto com o gráfico, estes valores indicam que a convergência de ambos os experimentos é bastante parecida. Isto ocorre mesmo com os dois experimentos sendo executados em redes com diferentes tamanhos e graus de dinamismo. Estes resultados indicam que, na média, um nodo qualquer tomado ao acaso em um instante de tempo qualquer conhece aproximadamente 99.5% da topologia real da rede. Por exemplo, se a rede possui 200 enlaces, aproximadamente 199 serão conhecidos pelo nodo.

Para confirmar a similaridade entre as convergências obtidas nos dois experimentos, foi realizado um teste-t [65]. Este é um teste estatístico que serve para medir a igualdade das médias de dois conjuntos de observações. Para isso, foram usadas as convergências médias de cada uma das 10 simulações de cada experimento, independente do tempo da medição. O teste-t resultou em uma forte evidência de que as médias obtidas para os dois

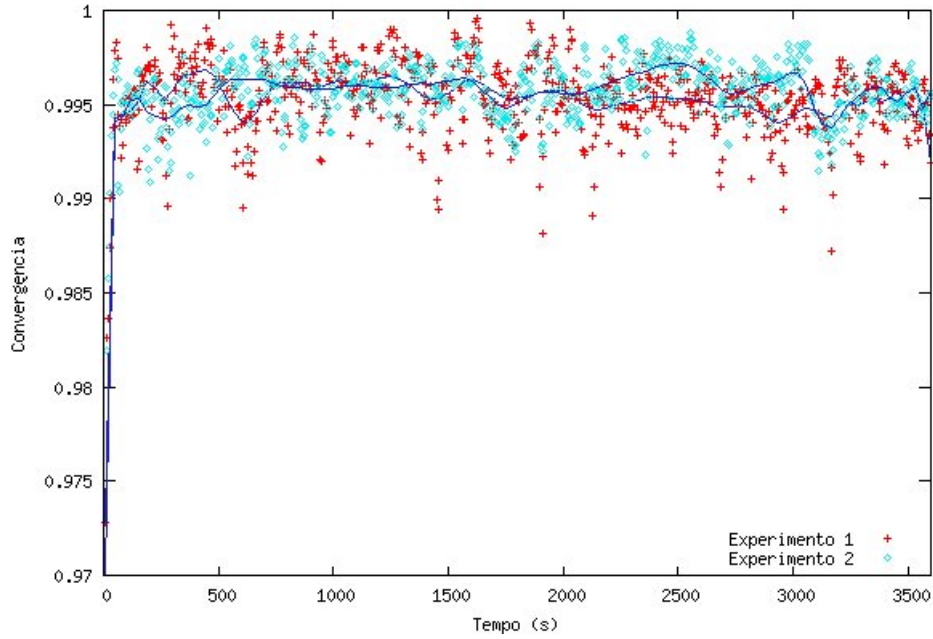


Figura 6.4: **Convergência média em 10 simulações sobre as redes padrão de cada conjunto.**

experimentos podem ser consideradas iguais. Isso confirma a hipótese de que a estratégia teve um desempenho similar nos dois experimentos, apesar das diferenças no tamanho das redes e no dinamismo.

A divergência média foi de 0.0053 para o experimento 1 e 0.0036 para o experimento 2. Para o experimento 1, ela variou entre 0 e 0.0205; e para o experimento 2 entre 0.0003 e 0.0127. Estes são valores baixos, e indicam que as descrições da topologia nos nodos normalmente contêm poucas informações errôneas, se comparadas ao número total de informações. A divergência média obtida no decorrer do tempo é mostrada no gráfico da figura 6.5.

Embora ambos os experimentos tenham apresentado uma divergência baixa, houve alguma diferença entre os valores médios obtidos. Essa diferença pode surgir em decorrência de diversos fatos. Uma provável causa para isso é que as redes do experimento 2 possuem mais nodos que as redes do experimento 1. Por isso, no experimento 2 os nodos tendem a ter uma maior quantidade de informações, o que diminui o impacto de uma informação errônea.

Outra possível causa para esta diferença nas divergências reside no fato de que os

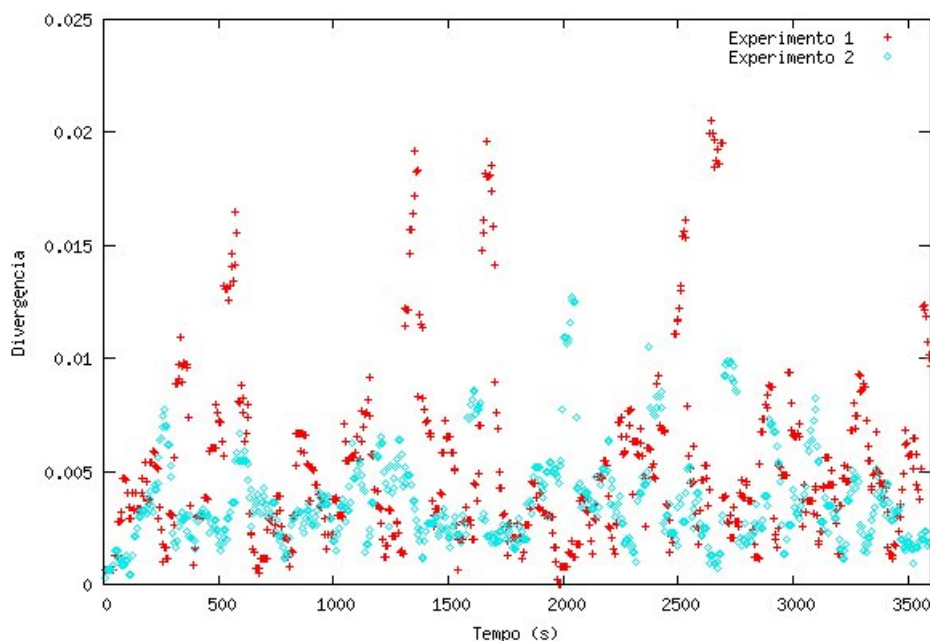


Figura 6.5: **Divergência média em 10 simulações sobre as redes padrão de cada conjunto.**

nodos removidos são diferentes para os dois experimentos. Pode-se observar que o impacto causado pela remoção de um nodo está diretamente ligado ao tamanho da sua vizinhança. Isso é esperado: se um nodo com 100 vizinhos é removido da rede, todos os nodos que conheçam a vizinhança do nodo removido passam a ter 100 informações errôneas em suas descrições. Por outro lado, se o nodo removido tem apenas 1 vizinho, o número de informações errôneas aumenta em apenas 1.

Para exemplificar essa situação, foi realizado um experimento baseado em 2 simulações. É dada uma rede inicial com 100 nodos. Após 1 minuto de simulação, um nodo é removido da rede. Em uma das simulações, foi removido o nodo com o maior número de vizinhos, e na outra foi removido o nodo com o menor número de vizinhos. Na primeira simulação, o nodo removido tinha 27 vizinhos, enquanto na segunda ele tinha apenas 1 vizinho. Havia um total de 146 enlaces na rede no momento da remoção dos nodos. A divergência na primeira simulação atingiu um valor máximo de 0.2012, enquanto na segunda simulação a divergência não passou de 0.0069. A figura 6.6 ilustra esse experimento.

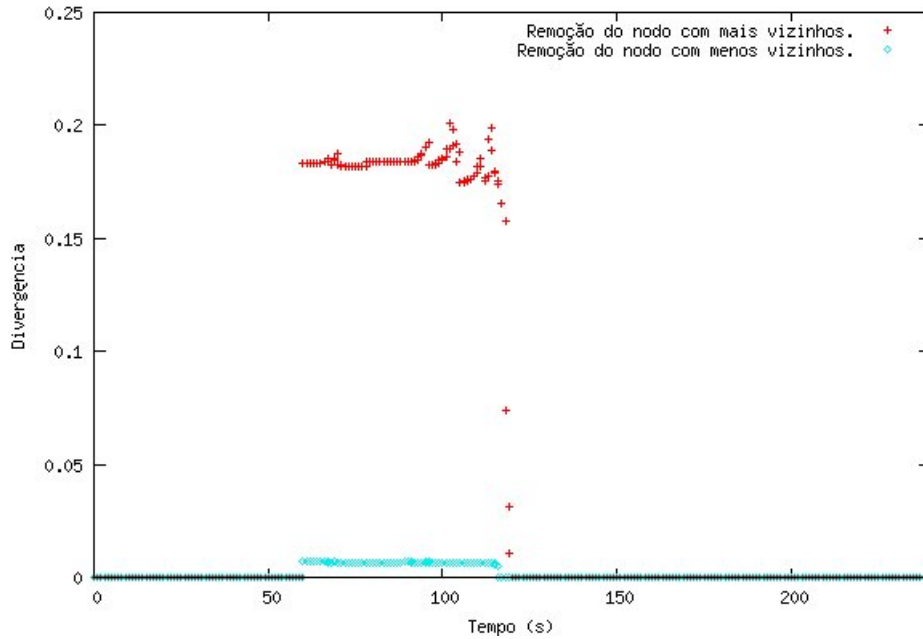


Figura 6.6: **Impacto da remoção de nodos com muitos ou poucos vizinhos.**

6.2.2 Dois Cenários Mais Complexos

Além dos experimentos apresentados na seção anterior, é importante que o desempenho da estratégia seja observado em algumas situações mais complexas. Para isso, foram definidos dois experimentos. No primeiro deles, são usadas as topologias iniciais das 10 redes padrão do primeiro conjunto (100 nodos). Porém, o dinamismo foi aumentado em 10 vezes. Para isso, os tempos máximos entre eventos foram definidos da seguinte forma: 30 segundos para a criação de um nodo, 1 minuto para a remoção de um nodo, 18 segundos para a criação de um enlace e 36 segundos para a remoção de um enlace. O segundo experimento faz uso dos arquivos de eventos das 10 redes do primeiro conjunto. Porém, as redes iniciais foram criadas com 500 nodos, ao invés de 100.

As simulações para estes experimentos foram executadas por apenas 10 minutos, com as métricas sendo apresentadas a cada 2 segundos. As convergências e divergências médias são mostradas nas figuras 6.7 e 6.8.

A convergência média obtida nas redes com mais dinamismo foi de 0.9819, e nas redes com 500 nodos foi de 0.9951. Já a divergência média foi de 0.0468 nas redes mais dinâmicas e de 0.0009 nas redes com 500 nodos. Junto com os gráficos, estes resultados sugerem

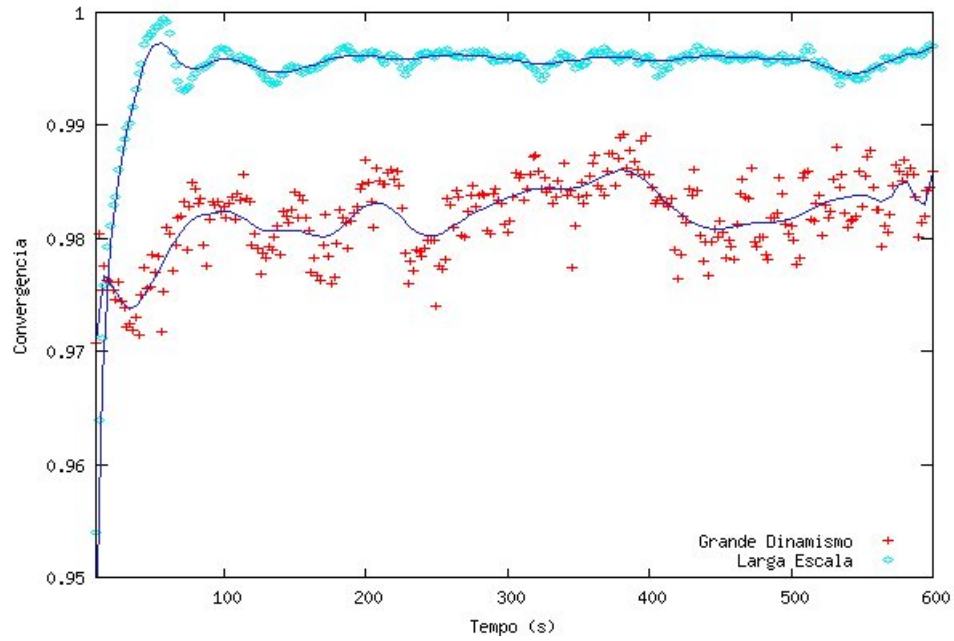


Figura 6.7: **Convergência média para redes muito dinâmicas e redes de larga escala.**

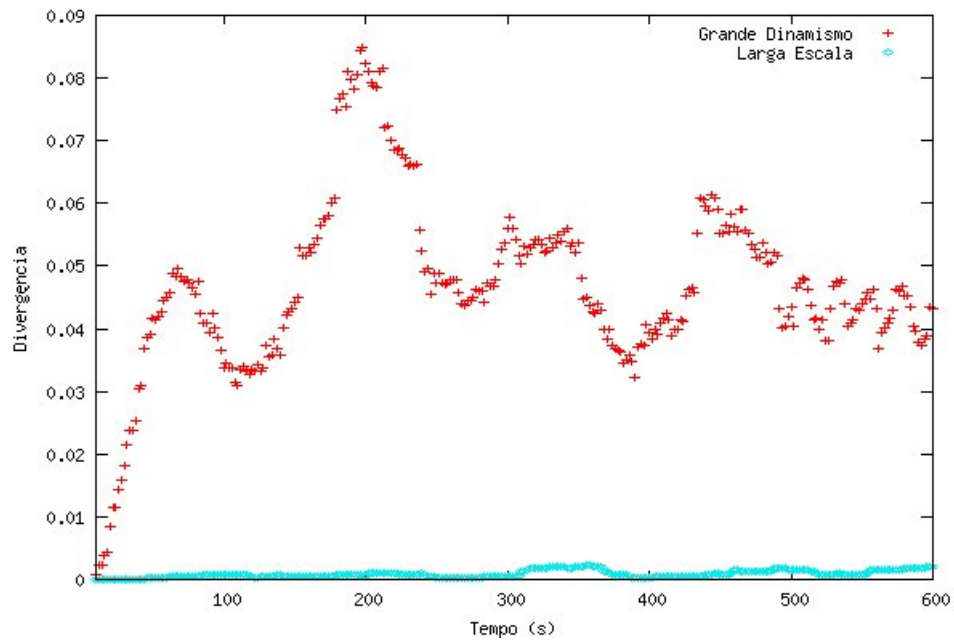


Figura 6.8: **Divergência média para redes muito dinâmicas e redes de larga escala.**

que a estratégia consegue se adaptar bem a um aumento no tamanho da rede. Observa-se que a convergência foi muito próxima daquela obtida nos experimentos apresentados na seção anterior, e a divergência, como esperado para uma rede com mais enlaces, foi

bastante reduzida. Já nas redes com mais dinamismo houve uma perda na convergência — que mesmo assim conseguiu manter um valor razoável — e um aumento na divergência média. Esse desempenho pior é esperado, já que um dinamismo maior implica em uma quantidade maior de informações desatualizadas.

Para confirmar as idéias apresentadas, foram feitos testes-t. Foram usadas nos testes as médias das convergências e divergências obtidas para cada experimento, independente do tempo. A convergência do experimento realizado nas redes mais dinâmicas foi comparada àquela obtida no experimento 1 da seção anterior. O teste resultou em uma forte evidência de que a média é menor para as redes mais dinâmicas. O teste também foi feito para as divergências, o que resultou em uma forte evidência de que a divergência média é maior para as redes mais dinâmicas. Por fim, a convergência do experimento realizado sobre as redes com mais nodos foi comparada à convergência do experimento 2 da seção anterior. O teste resultou em uma forte evidência de que as médias são iguais para ambos os experimentos.

Pode-se então concluir que um aumento no dinamismo da rede resulta em descrições com uma qualidade mais baixa: convergência menor e divergência maior. Observa-se que a redução na convergência talvez pudesse ser resolvida com um ajuste nos parâmetros da estratégia, por exemplo, reduzindo o TtM. Já a divergência só poderia ser reduzida com modificações na estratégia em si, por exemplo, acrescentando às descrições informações sobre enlaces que deixaram de existir. A estratégia se adaptou bem ao aumento de tamanho da rede. Observou-se que isso ocorre com um aumento no número de agentes usado, como é mostrado mais adiante, na seção 6.2.5.

6.2.3 Idade e Envelhecimento das Informações

Como dito na seção 5.8, a idade máxima que uma informação pode ter antes de ser descartada deve ser definida com cautela. Se esta idade for muito alta, um grande número de informações desatualizadas permanecerá na rede, aumentando assim a divergência das descrições. Por outro lado, se ela for muito baixa, várias informações podem ser descartadas pouco tempo após terem sido recebidas por um nodo — ou mesmo antes

disso — o que pode exigir um aumento no tráfego de agentes na rede. Além disso, quanto maior for a idade máxima para as informações, maior será a quantidade de informações presentes nos nodos e carregadas pelos agentes, e mais recursos serão consumidos nos nodos e na rede. Para avaliar o impacto dessa idade máxima na qualidade das descrições produzidas, foram realizados quatro experimentos.

O primeiro experimento foi feito sobre uma das redes padrão do primeiro conjunto, com $n = 100$, $m_0 = 5$, $p = 1$, $\beta = 0$, e o arquivo de eventos associado. Foram executadas 5 simulações, com diferentes valores para a idade máxima das informações nos nodos. As idades usadas foram: 15 s, 30 s, 60 s (o valor padrão), 90 s e 120 s. Cada um destes valores foi usado em uma simulação, com duração total de 20 minutos, e com os resultados sendo apresentados a cada 2 segundos. Todas as simulações foram executadas com uma mesma semente para o gerador de números aleatórios. Pela forma como o ANTD foi implementado, isso faz com que até mesmo os eventos, o número de agentes e os caminhos seguidos sejam iguais em todas as simulações. Ou seja, a única diferença entre as simulações está na idade das informações descartadas.

A figura 6.9 mostra a convergência obtida para cada simulação, exceto aquela na qual a idade máxima foi de 120 s. Esta simulação foi removida do gráfico porque seus resultados foram muito próximos aos da simulação na qual a idade máxima era de 90 s.

A convergência média produzida para cada simulação foi: 0.5045 quando a idade máxima é 15 s; 0.8396 quando a idade máxima é 30 s; 0.9829 quando a idade máxima é 60 s; 0.9950 quando a idade máxima é 90 s; e 0.9965 quando a idade máxima é 120 s. Pode-se observar que, em todos os casos, um aumento na idade máxima resultou em uma melhora na convergência, embora este ganho tenha diminuído para as maiores idades.

A figura 6.10 mostra as divergências médias obtidas nesse experimento para as simulações com idade máxima igual a 15 s e 120 s. Pode-se observar que os aumentos na divergência ocorrem ao mesmo tempo, mas quando a idade máxima é 120 s, a divergência tende a ser maior e a permanecer elevada por mais tempo. As demais simulações tiveram um comportamento similar, com a divergência aumentando em tempo e valor conforme se aumenta a idade máxima das informações.

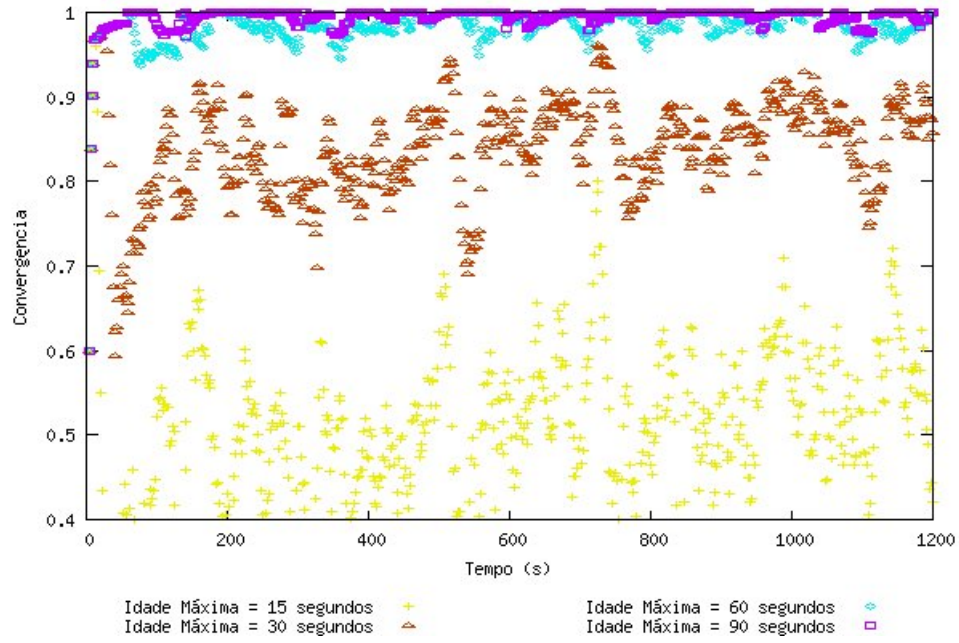


Figura 6.9: Convergência da descrição com diferentes idades máximas para as informações nos nodos.

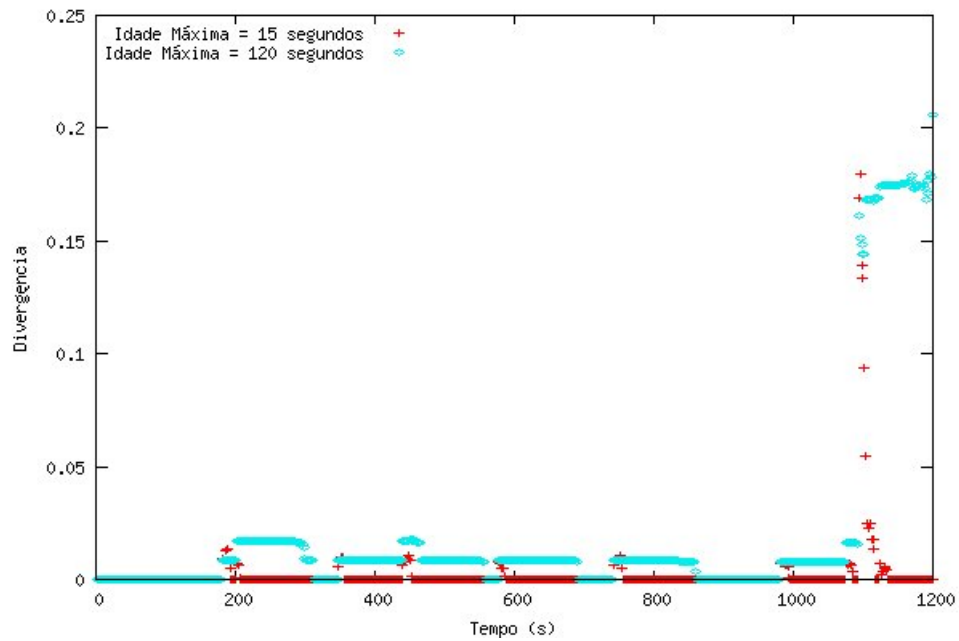


Figura 6.10: Divergência da descrição com diferentes idades máximas para as informações nos nodos.

Na simulação com idade máxima de 15 s, a divergência média foi de 0.0021. Para as outras idades máximas, 30 s, 60 s, 90 s e 120 s, a divergência média foi, respectivamente, 0.0053, 0.01137, 0.01718 e 0.02144. Com base nesses valores e no gráfico, pode-se concluir

que há um aumento na divergência quando se aumenta a idade máxima das informações. Este é um resultado esperado, já que o aumento na idade máxima das informações faz com que informações errôneas permaneçam por mais tempo nas descrições.

O primeiro experimento foi então repetido, mas variando a idade máxima das informações carregadas pelos agentes de descoberta. Os valores usados foram os mesmos do experimento anterior. As convergências médias para as idades máximas de 15 s, 30 s, e 60 s foram, respectivamente, 0.9595, 0.9804 e 0.9829. As divergências médias para essas mesmas simulações foram de 0.0101, 0.0111 e 0.0114. As descrições obtidas com as idades máximas de 90 s e 120 s foram idênticas àquelas obtidas com 60 s, tendo assim as mesmas convergências e divergências.

Os resultados obtidos no segundo experimento confirmam a hipótese de que um aumento na idade das informações que são descartadas leva a um aumento na convergência e na divergência das descrições. Porém, o impacto da variação na idade máxima foi menor no caso das informações carregadas pelos agentes. Isso indica que os agentes estão conseguindo levar cada informação a um grande número de nodos antes do seu descarte. Por isso, quando a idade máxima das informações carregadas aumenta, em muitos casos as informações mais antigas já estão presentes nas descrições de todos os nodos visitados, por terem sido trazidas por outros agentes ou pelo mesmo agente em um momento anterior.

Para confirmar os resultados dos dois experimentos anteriores, eles foram repetidos em uma das redes padrão do segundo conjunto, com 200 nodos na topologia inicial. Os resultados obtidos foram bastante parecidos com os dos experimentos anteriores. A convergência foi ligeiramente mais baixa quando se consideraram os tempos mais baixos, mas chegou a um patamar próximo quando as idades máximas das informações (tanto aquelas presentes nos nodos quanto para aquelas carregadas por agentes) foi igual ou superior a 60 s. A divergência foi bem menor nestes casos, mas como dito na seção 6.2.1, é esperado que redes maiores tenham descrições com uma divergência menor.

O que deve ser destacado nos resultados do terceiro e quarto experimentos é que a relação entre as diferentes combinações de parâmetros permaneceu inalterada. Ou seja, aumentos na idade máxima das informações levaram a aumentos na convergência e na

divergência. Para as informações mantidas nas descrições da topologia, este aumento se manteve até a maior idade experimentada (120 s). Já para as informações carregadas pelos agentes, não houveram mudanças para idades máximas superiores a 60 s.

6.2.4 Migração

Este trabalho propõe dois métodos para a escolha do próximo destino de um agente. Em um deles, um dos vizinhos do nodo é selecionado aleatoriamente. No outro método, mais sofisticado, os agentes depositam feromônios associados aos enlaces percorridos. Esses feromônios são usados para que os agentes procurem “fugir” uns dos outros, se espalhando pela rede de uma maneira mais eficiente.

Para verificar se a seleção baseada em feromônios produz resultados melhores que a seleção aleatória, o primeiro experimento da seção 6.2.1 foi repetido, com o método de seleção de destino aleatório sendo usado. Os gráficos das figuras 6.11, 6.12 e 6.13 mostram, respectivamente, as convergências médias, divergências médias e as medianas do número de agentes obtidas para ambos os métodos de seleção de destino.

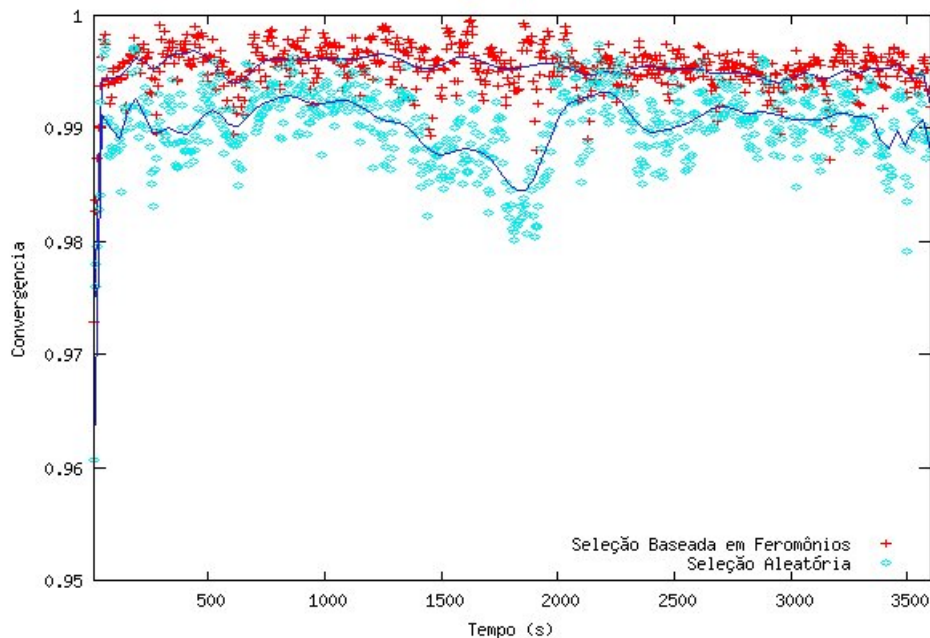


Figura 6.11: Convergência média da descrição nas redes do primeiro conjunto, com seleção de destino aleatória e baseada em feromônios.

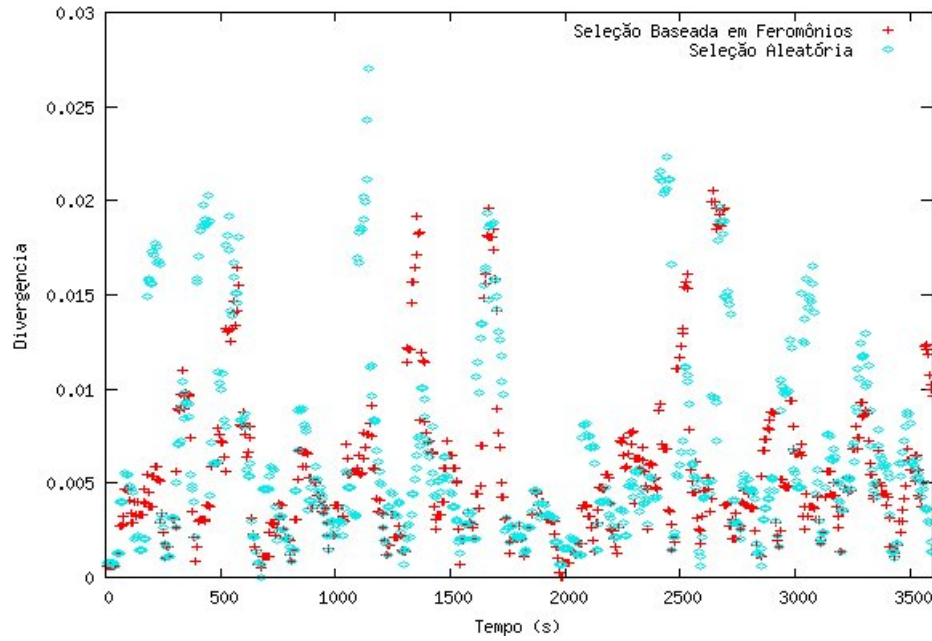


Figura 6.12: Divergência média da descrição nas redes do primeiro conjunto, com seleção de destino aleatória e baseada em feromônios.

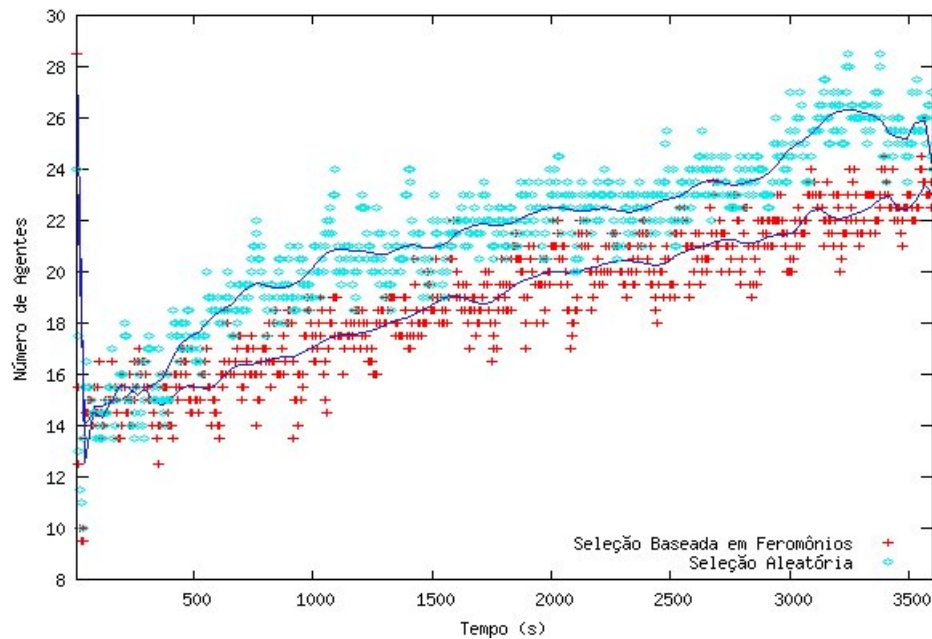


Figura 6.13: Mediana do número de agentes usados nas redes do primeiro conjunto, com seleção de destino aleatória e baseada em feromônios.

A convergência média foi de 0.9903 para a seleção aleatória e de 0.9953 para a seleção baseada em feromônios. Apesar de ser uma diferença pequena, o gráfico indica que a seleção baseada em feromônios teve um desempenho melhor no decorrer do tempo. O

mesmo pode ser dito para o número de agentes: a mediana foi de 22 para a seleção aleatória e de 19 para a seleção baseada em feromônios. Novamente, apesar de ser uma diferença pequena, existe no gráfico uma indicação de que a seleção baseada em feromônios fez uso de um número menor de agentes.

A divergência média foi de 0.0064 para a seleção aleatória e de 0.0053 para a seleção baseada em feromônios. Porém, neste caso o gráfico não dá qualquer indicação de que a seleção baseada em feromônios produza descrições com menor convergência. De fato, uma possível explicação para esta diferença está no número médio de enlaces da rede: 137.52 para as redes usadas na seleção aleatória e 142.32 para as redes usadas na seleção baseada em feromônios. Como dito na seção 6.2.1, uma rede mais conectada tende a ter descrições com uma divergência menor.

O segundo experimento da seção 6.2.1 também foi repetido com o método aleatório para a seleção de destino. Os gráficos para a convergência média, divergência média e mediana do número de agentes são mostrados, respectivamente, nas figuras 6.14, 6.15 e 6.16.

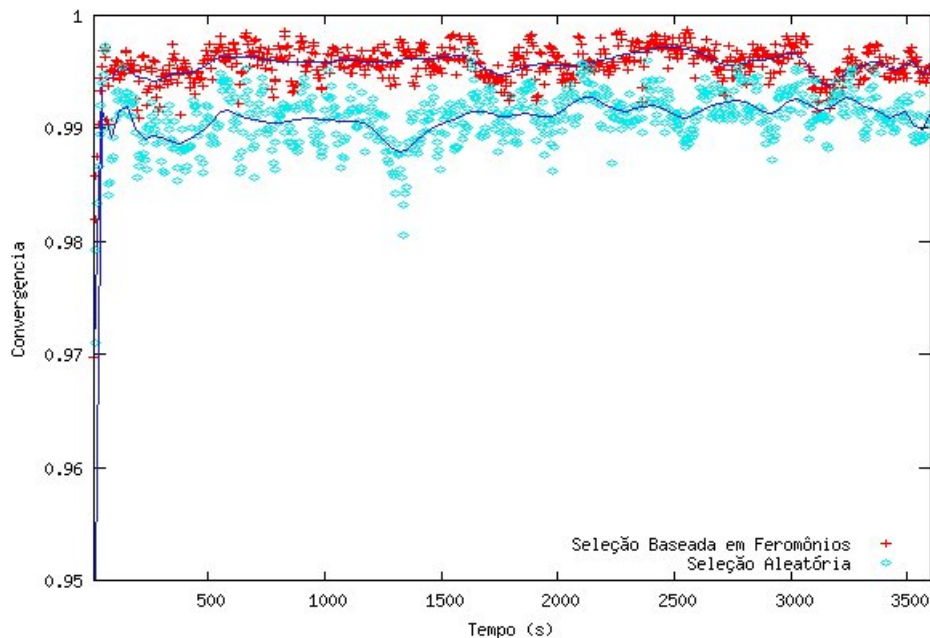


Figura 6.14: **Convergência média da descrição nas redes do segundo conjunto, com seleção de destino aleatória e baseada em feromônios.**

A convergência média para a seleção aleatória foi de 0.9909, e para seleção baseada em

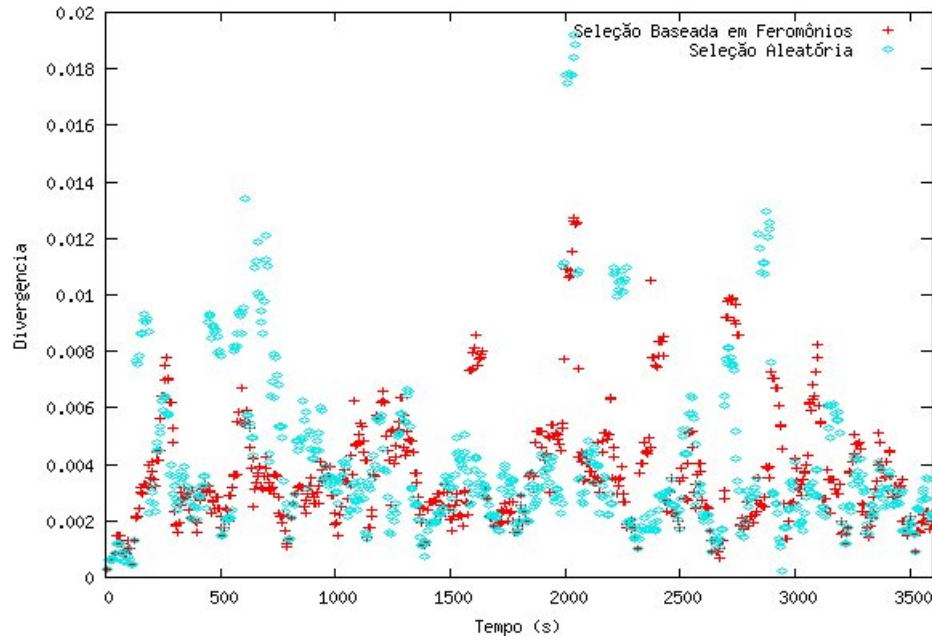


Figura 6.15: Divergência média da descrição nas redes do segundo conjunto, com seleção de destino aleatória e baseada em feromônios.

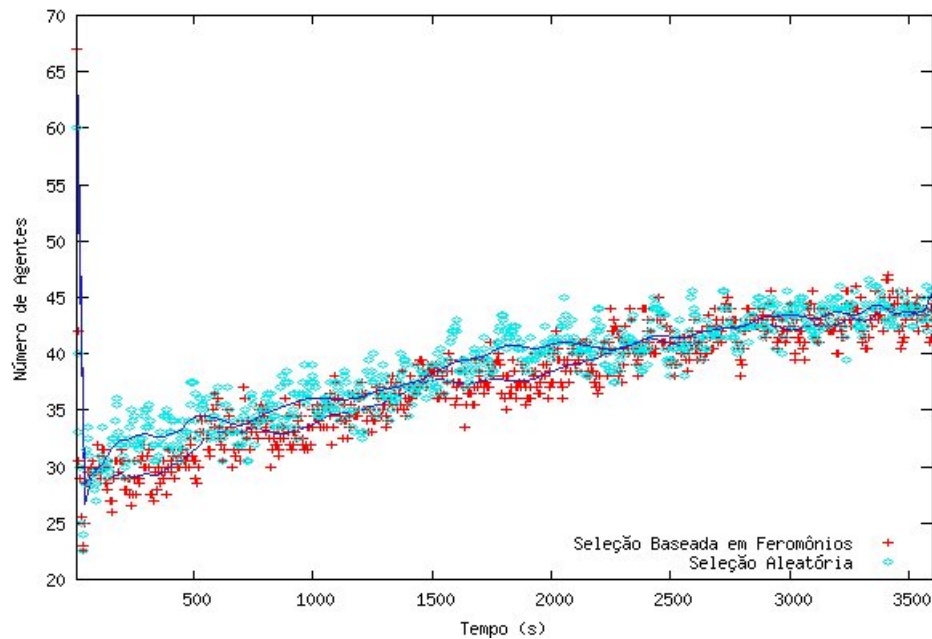


Figura 6.16: Mediana do número de agentes usados nas redes do segundo conjunto, com seleção de destino aleatória e baseada em feromônios.

feromônios foi de 0.9957. Tanto estes resultados quanto o gráfico são bastante similares aos do experimento anterior. A mediana da população também foi ligeiramente menor para a seleção baseada em feromônios: 38, contra 39.5 para a seleção aleatória. Porém,

esta vantagem não é evidente no gráfico da população de agentes. A divergência média para os dois métodos foi bastante próxima: 0.004 e 0.0036, respectivamente, para a seleção aleatória e para a seleção baseada em feromônios.

A partir desses resultados, algumas hipóteses podem ser formuladas. A primeira diz que o número de agentes usados não está relacionado ao método usado para a seleção de destinos, com o mesmo podendo ser dito a respeito da divergência. A segunda diz respeito à convergência: o método de seleção de destinos baseado em feromônios produz descrições com convergência ligeiramente maior que o método aleatório.

Para confirmar a idéia de que o número de agentes não tem relação com o método usado para a seleção de destinos, foi realizado, para cada um dos dois experimentos, um teste-t sobre a mediana do número de agentes. Foram usadas as medianas de cada uma das simulações, independente do tempo. Ambos os testes indicaram que a média dessas medianas é igual para os dois métodos de seleção de destino.

O número médio de migrações de agentes por segundo também foi bastante parecido para os dois métodos. No primeiro experimento, este número foi de 34.26 para a seleção aleatória e 34.15 para a seleção baseada em feromônios. No segundo experimento, os resultados foram, respectivamente, 70.61 e 70.94. Estes valores sugerem que não existe relação entre o número de agentes e o método usado para a seleção de destinos, e que o comportamento mostrado na figura 6.13 é um caso isolado.

Para confirmar a segunda hipótese, de que a seleção baseada em feromônios gera descrições com uma convergência maior, o teste-t também foi usado. A convergência média independente do tempo de cada uma das simulações foi calculada e usada como entrada para o teste. Foram feitos dois testes, um para cada experimento, usando as simulações correspondentes. Ambos os testes resultaram em uma forte evidência de que existe uma diferença entre as convergências obtidas com cada método de seleção de destinos. Eles também apontam que a média obtida quando se usa a seleção baseada em feromônios é maior, confirmando a hipótese.

Para ilustrar mais claramente essa hipótese, foi realizado mais um experimento. Foram geradas 10 redes estáticas com 500 nodos e um número de enlaces variando entre 555 e

1110 (a média foi de 744 enlaces). Cada uma dessas redes foi usada em duas simulações de 40 segundos, uma para cada método de seleção de destinos. Cada simulação assumiu uma população inicial de 100 agentes de descoberta, e o ajuste automático para a população de agentes não foi usado, de forma que o número de agentes é fixo e igual para ambos os métodos. A figura 6.17 mostra a convergência média obtida.

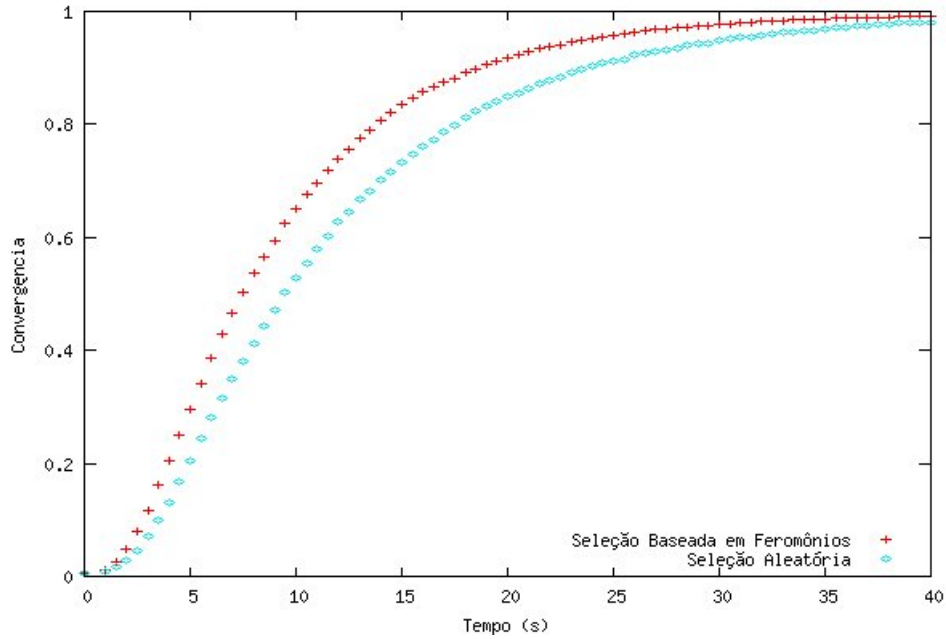


Figura 6.17: Convergência média em redes de 500 nodos, partindo de um estado no qual não há informações nas descrições.

Pode-se observar que, partindo de um estado no qual não há informações nas descrições da topologia, a convergência aumentou mais rapidamente quando a seleção baseada em feromônios foi usada. Essa mesma vantagem foi constatada quando se considerou cada par de simulações isoladamente, ou seja, em todas as 10 redes, a convergência aumentou mais rapidamente para a seleção baseada em feromônios. Isso indica que os agentes estão se espalhando pela rede de maneira mais eficiente, produzindo descrições com uma convergência mais alta — o que confirma a hipótese levantada.

6.2.5 População de Agentes de Descoberta

A estratégia proposta por este trabalho inclui uma abordagem para que a população de agentes de descoberta se adapte automaticamente a mudanças no tamanho da rede. É

preciso então avaliar se o número de agentes gerado por esta abordagem é o suficiente para que a descoberta da topologia ocorra com eficiência em diferentes cenários.

Vários dos experimentos realizados em seções anteriores já demonstram de certa forma o ajuste da população à rede. Por exemplo, nos experimentos gerais (seção 6.2.1), foram usadas redes de diferentes tamanhos e graus de dinamismo. A convergência obtida foi bastante similar para as redes do primeiro e segundo conjuntos. Observa-se que a mediana do número de agentes usados foi diferente para os dois experimentos: 19 agentes no primeiro experimento e 38 no segundo. Ou seja, o segundo experimento usou aproximadamente o dobro de agentes do primeiro — resultado esperado, já que as redes do segundo conjunto têm inicialmente o dobro do tamanho das redes do primeiro conjunto. Já no experimento feito na rede de 500 nodos apresentado na seção 6.2.2, a mediana foi de 77 agentes.

Outra observação que se pode fazer sobre os experimentos da seção 6.2.1 é que eles são realizados sobre redes cujo tamanho tende a apresentar um crescimento com o passar do tempo. A figura 6.18 mostra o número médio de nodos e enlaces e a mediana do número de agentes usados no primeiro experimento. Pode-se observar que existe um pequeno aumento no tamanho médio da rede com o passar do tempo, e que esse aumento é acompanhado por um aumento na população de agentes. O segundo experimento apresentou um comportamento bastante parecido com este.

Também se constatou que no início das simulações o número de agentes de descoberta é bastante grande, estando próximo ao número de nodos. Isso ocorre porque um nodo que se conecta à rede não tem qualquer concentração de feromônios. Por este motivo, todos os nodos, no momento em que se conectam à rede, criam um agente. Como no início das simulações pode-se considerar que todos os nodos estão se conectando à rede, todos criarão agentes. Após alguns segundos, os nodos detectam o excesso de agentes e acabam levando a sua população a um patamar mais apropriado. Este aparente “excesso” de agentes no início das simulações faz com que a convergência aumente rapidamente nos nodos, chegando ao seu patamar médio em poucos segundos. Este comportamento foi observado em todos os outros experimentos realizados.

Esses experimentos também podem ser usados para reforçar uma afirmação feita na

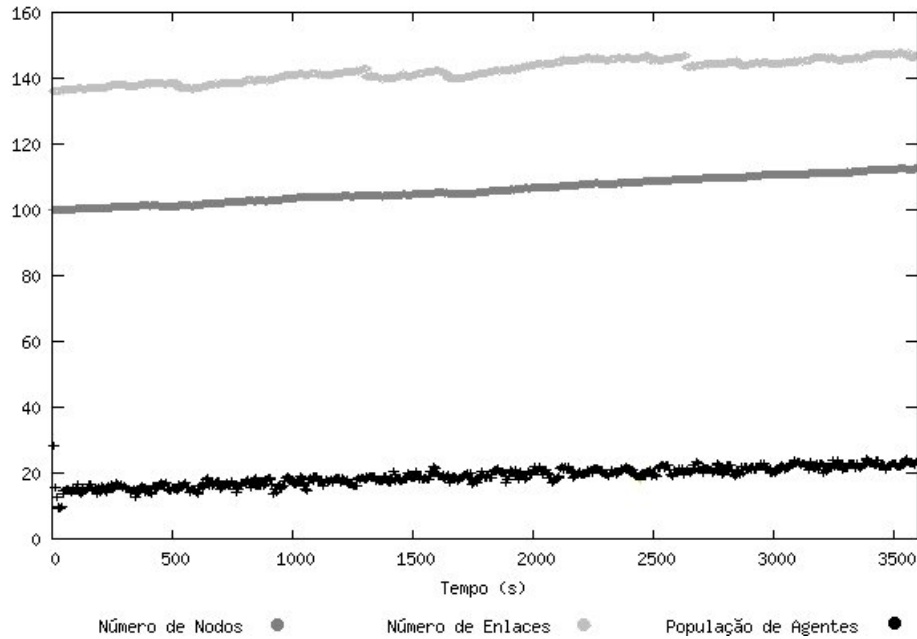


Figura 6.18: Número médio de nodos e enlaces e mediana do número de agentes com o passar do tempo, nas redes do primeiro conjunto.

seção 5.6. Naquela seção, foi dito que a população de agentes de descoberta tende a ser maior em redes mais conectadas.

Para observar isso, foram tomadas, para cada uma das simulações, as seguintes medidas, independentes do tempo: média do número de nodos, média do número de enlaces e mediana da população de agentes. Foi então calculado o coeficiente correlação entre o número de nodos e a população, e entre o número de enlaces e a população. Para o primeiro experimento, o coeficiente foi de -0.4742 entre o número de nodos e a população, e de 0.8805 entre o número de enlaces e a população. Para o segundo experimento, os coeficientes foram, respectivamente, 0.06366 e 0.9361. Estes resultados indicam que existe uma ligação bastante forte entre o número de enlaces da rede e o número de agentes — e que esta relação é mais forte que aquela existente entre o número de nodos e o número de agentes.

Para ilustrar melhor como a população de agentes de descoberta se ajusta a uma rede dinâmica, foram realizadas 10 simulações usando redes iniciais de apenas 10 nodos. Após 1 minuto de simulação, o tamanho da rede cresce abruptamente, com 120 nodos sendo conectados à rede num intervalo de 20 minutos. Então, após 10 minutos, 60 desses nodos

são removidos da rede, um a cada 10 segundos. A simulação é ainda executada por mais 9 minutos, nos quais nenhum evento ocorre. Cada simulação foi então repetida com uma população estática de 16 agentes de descoberta, que se espalham uniformemente pela rede no início da simulação. O gráfico da figura 6.19 mostra o número médio de nodos e a mediana da população de agentes com o passar do tempo.

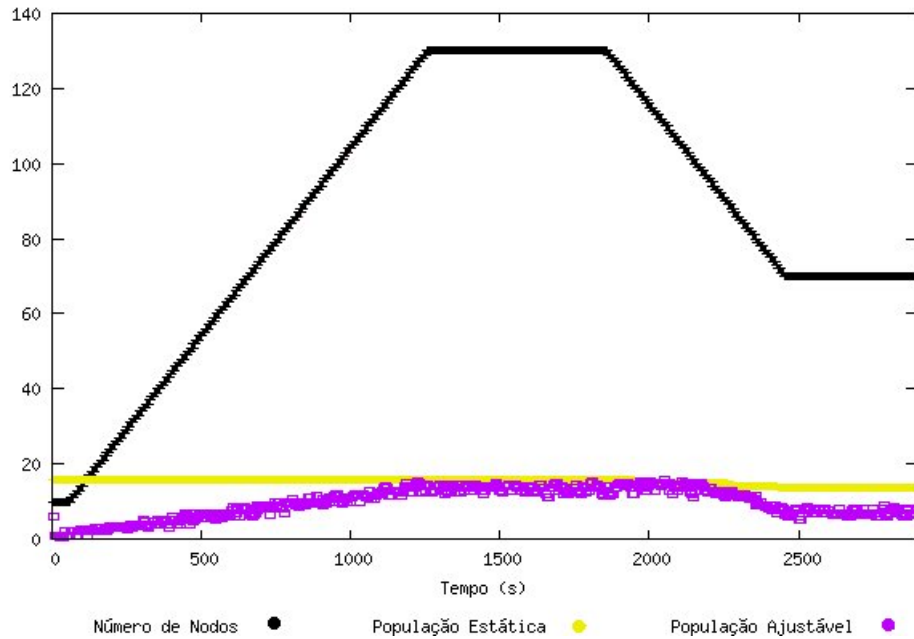


Figura 6.19: **Número de nodos e tamanho da população de agentes com o passar do tempo.**

A convergência média obtida quando a população ajustável foi usada foi de 0.9573, contra 0.962 da população estática. A população dinâmica teve uma mediana de 10 agentes, com uma média de 20.28 migrações por segundo tendo ocorrido durante a simulação. Quando a população estática foi usada, ocorreram em média 33.88 migrações por segundo. Estes números indicam que a população consegue se ajustar bem às mudanças na rede, reduzindo o tráfego de agentes quando os mesmos não se fazem necessários. Deve-se destacar que cada nodo faz esse ajuste com base apenas em informações locais — as concentrações de feromônio e o tamanho da sua própria vizinhança.

Dois fatos curiosos puderam ser observados nos resultados desse experimento. Um deles foi que a população que devia ser estática foi ligeiramente reduzida quando nodos foram removidos da rede. Isso ocorre porque um nodo que é removido da rede também destrói

quaisquer agentes que porventura estivessem nele. O outro fato curioso foi que a remoção dos nodos na rede fez a população ajustável de agentes sofrer um pequeno aumento por alguns instantes. Isso ocorre porque a remoção de nodos faz com que as concentrações dos feromônios associados aos enlaces desses nodos sejam reduzidas rapidamente, o que faz com que os outros nodos criem mais agentes.

6.2.6 Tempo Antes de Migrar

Para analisar o papel do Tempo Antes de Migrar (TtM — *Time to Migrate*, descrito na seção 5.7) na estratégia proposta, foi realizado um experimento no qual o seu valor foi variado. Quatro simulações foram feitas sobre cada uma das 10 redes do primeiro conjunto. Em cada uma das simulações, o TtM foi diferente. Foram usados os seguintes tempos: 0 ms, 200 ms (o valor padrão), 1 s e 2 s. As simulações tiveram duração de 20 minutos, com as métricas sendo apresentadas a cada 2 s. As figuras 6.20 e 6.21 mostram a convergência média e a mediana do número de agentes no decorrer do tempo.

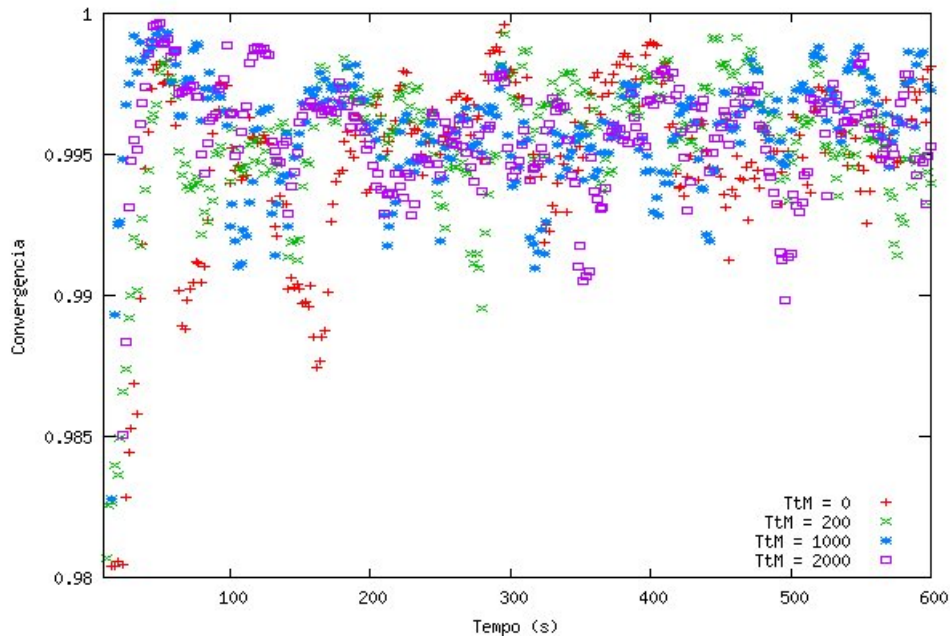


Figura 6.20: **Convergência da descrição com diferentes Tempos Antes de Migrar.**

A tabela 6.3 mostra algumas das métricas obtidas, independentes do tempo.

Pode-se observar que a convergência obtida foi bastante próxima em todos os casos.

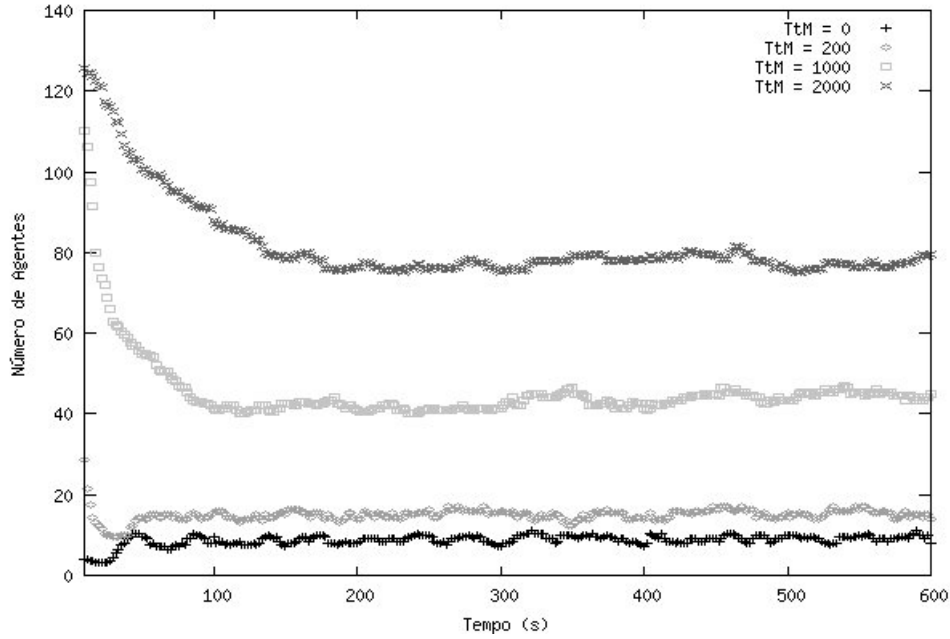


Figura 6.21: População de agentes com diferentes Tempos Antes de Migrar.

	TtM = 0	TtM = 200	TtM = 1000	TtM = 2000
Convergência Média	0.9954	0.9955	0.9957	0.9948
Divergência Média	0.0048	0.0047	0.0055	0.0061
Mediana do Número de Agentes	9.5	16	45.5	80
Média de Migrações / s em Cada Nodo	0.3252	0.333	0.3504	0.3548

Tabela 6.3: Resultados obtidos com diferentes Tempos Antes de Migrar.

Para confirmar a sua igualdade, foram calculadas as convergências médias para cada simulação, independente do tempo. Foram então realizados testes-t, usando como entrada os resultados para cada um dos valores de TtM e aqueles obtidos com o TtM = 0. Os testes apontaram uma forte evidência de que as médias foram iguais para todos os experimentos.

Já com relação às divergências, pode-se observar um aumento quando os maiores Tempos Antes de Migrar foram usados. Foram feitos testes-t para analisar se essa variação foi significativa. Os testes indicaram que as médias das divergências podem ser consideradas iguais para todos os valores de TtM, apesar da diferença encontrada.

Quanto à população de agentes, é evidente o aumento no número de agentes quando o TtM aumenta. Esse é um comportamento esperado, que demonstra a estratégia usada para o ajuste da população de agentes em funcionamento. Ou seja, se o TtM aumenta, os agentes visitam um número menor de nodos. Isso faz com que os nodos “compensem”

a diferença aumentando o número de agentes na rede.

Por fim, o número médio de migrações por segundo para cada nodo parece sofrer um aumento quando o TtM aumenta. Testes-t foram feitos, usando os resultados de cada experimento para cada valor de TtM. Os testes apontaram igualdade apenas nos resultados obtidos com TtM igual a 1000 e a 2000. Nos demais casos, todos os testes indicaram que a média do número de migrações é maior para os maiores valores de TtM.

Esses resultados mostram que neste experimento as perspectivas que existiam sobre o Tempo Antes de Migrar não se confirmaram. Isso provavelmente ocorreu por causa do ajuste automático da população de agentes. Os nodos compensam a redução no TtM com a destruição de agentes. Por isso, o fluxo de agentes não foi reduzido quando o TtM foi aumentado — pelo contrário, o número de migrações aumentou. Também não foi encontrada evidência de que um número maior de agentes realizando mais migrações produz descrições com uma qualidade melhor que um número menor de agentes realizando mais migrações. Todos esses resultados indicam que o TtM não melhora a qualidade da estratégia como um todo, caso o ajuste automático da população de agentes esteja sendo usado.

6.2.7 Conclusões dos Experimentos

A partir dos resultados dos experimentos apresentados nas subseções anteriores, chegou-se a várias conclusões. Elas são aqui listadas, de forma resumida.

Em primeiro lugar, observou-se que a estratégia proposta produz descrições bastante similares à topologia real da rede, com uma convergência alta (média superior a 0.995) e uma divergência baixa (média inferior a 0.06). Há uma ligeira perda de desempenho quando o dinamismo da rede aumenta. Já quando há mudanças no tamanho da rede, a estratégia consegue ajustar automaticamente a população de agentes de forma a não sofrer perdas de desempenho.

Foi constatado que um aumento na idade máxima das informações que são mantidas nos nodos ou carregadas pelos agentes leva a um aumento na convergência e na divergência das descrições obtidas. Também foi mostrado que a seleção de destinos baseada

em feromônios consegue produzir resultados melhores que a seleção aleatória. Por fim, observou-se que o uso do Tempo Antes de Migrar não trouxe a redução esperada no tráfego de agentes, pois o método proposto para o ajuste da população compensa o aumento no número de migrações de cada agente com uma redução no seu número. O uso do TtM também não melhorou a qualidade das descrições da topologia obtidas.

Capítulo 7

Conclusão

Neste trabalho, foi proposta uma estratégia baseada em agentes móveis inteligentes para a descoberta da topologia de redes dinâmicas e descentralizadas, tais como redes *peer-to-peer* descentralizadas não estruturadas e redes móveis sem fio ad hoc. A estratégia proposta foi avaliada através de uma série de experimentos, que demonstraram que as descrições da topologia produzidas possuem um alto grau de similaridade com a topologia real da rede. Outros resultados indicaram que o método baseado em feromônios para a seleção de destinos dos agentes produz melhores resultados que a seleção aleatória, e que a população de agentes pode ser dinamicamente adaptada à rede, conforme esta muda de tamanho.

Deve-se observar que a estratégia proposta foi desenvolvida especialmente para redes com alto grau de dinamismo, e com um número grande de nodos. Em redes menores ou menos dinâmicas, estratégias baseadas em eventos, como a inundação da rede com mensagens que avisam os nodos sobre mudanças na topologia, podem ter um desempenho melhor e com um *overhead* menor. Portanto, uma comparação entre a estratégia proposta e outras abordagens deve ser feita no futuro.

Outros trabalhos futuros incluem modificações na estratégia para que os agentes disseminem informações a respeito de nodos ou enlaces que deixam de existir. Esse tipo de modificação poderia ser usado, por exemplo, para especializar a estratégia para o diagnóstico do estado dos nodos. Outra modificação possível seria a criação de novos algoritmos para a seleção de destinos, baseados nas informações já coletadas pelos agentes, que po-

dem ser encontradas nas descrições da topologia nos nodos. Além disso, o envelhecimento das informações pode ser modificado para funcionar com base em contadores, para que a estratégia seja usada em redes assíncronas. Por fim, o simulador desenvolvido pode ser estendido, de forma a incluir outros tipos de cenários para a simulação, como redes ad hoc ou redes móveis híbridas.

Referências Bibliográficas

- [1] Joy Algesheimer, Christian Cachin, Jan Camenisch, e Günter Karjoth. Cryptographic Security for Mobile Code. *Proceedings of the 2001 IEEE Symposium on Security and Privacy (SP'2001)*. IEEE Computer Society, 2001.
- [2] Alasdair Allan, Tim Naylor, Iain Steele, Dave Carter, Tim Jenness, Frossie Economou, e Andy Adamson. eSTAR: Astronomers, Agents, and when Robotic Telescopes Aren't... *Astronomical Data Analysis Software and Systems XIII (ADASS XIII)*, volume 314, Strasbourg, France, 2003.
- [3] Kaizar A. Amin. Resource Efficient and Scalable Routing Using Intelligent Mobile Agents. Dissertação de Mestrado, University of North Texas, maio de 2003.
- [4] Kaizar A. Amin, John T. Mayes, e Armin R. Mikler. Agent-Based Distance Vector Routing. *Proceedings of the Third International Workshop on Mobile Agents for Telecommunication Applications (MATA'2001)*, pág. 41–50. Springer-Verlag, 2001.
- [5] Payman Arabshahi e Andrew Gray et. al. Adaptive Routing in Wireless Communication Networks Using Swarm Intelligence. *Proceedings of the 19th AIAA International Communications Satellite Systems Conference*, Toulouse, France, 2001.
- [6] Yigal Bejerano, Yuri Breitbart, Minos N. Garofalakis, e Rajeev Rastogi. Physical Topology Discovery for Large Multi-Subnet Networks. *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'2003)*, California, USA, 2003.
- [7] Andrzej Bieszczad, Bernard Pagurek, e Tony White. Mobile Agents for Network Management. *IEEE Communications Surveys and Tutorials*, 1(1), 1998.
- [8] Chatschik Bisdikian. An Overview of the Bluetooth Wireless Technology. *IEEE Communication Magazine*, pág. 86–94, Dezembro de 2001.
- [9] Eric Bonabeau, Marco Dorigo, e Guy Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press Inc., 1999.
- [10] Bot Spot — The Spot for All Bots. Disponível em <<http://www.botspot.com>>. Acessado em Janeiro de 2005.
- [11] Peter Braun, Christian Erfurth, e Wilhelm R. Rossak. An Introduction to the Tracy Mobile Agent System. Relatório Técnico 2000/24, Friedrich-Schiller-Universität Jena, Institut für Informatik, Setembro de 2000.

- [12] Tian Bu e Donald F. Towsley. On Distinguishing between Internet Power Law Topology Generators. *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'2002)*, 2002.
- [13] Giacomo Cabri, Letizia Leonardi, e Franco Zambonelli. Coordination in Mobile Agent Applications. Relatório Técnico DSI-97-24, Dipartimento di Scienze dell'Ingegneria, Università di Modena, Outubro de 1997.
- [14] Giacomo Cabri, Letizia Leonardi, e Franco Zambonelli. How to Coordinate Internet Applications based on Mobile Agents. *Proceedings of the 7th Workshop on Enabling Technologies (WETICE'98)*, pág. 104–109. IEEE Computer Society, 1998.
- [15] Giacomo Cabri, Letizia Leonardi, e Franco Zambonelli. Mobile Agent Technology: Current Trends and Perspectives. *Congresso Annual AICA '98*, Napoli, Italy, 1998.
- [16] Gianni Di Caro e Marco Dorigo. AntNet: Distributed Stigmergetic Control for Communications Networks. *Journal of Artificial Intelligence Research*, 9:317–365, 1998.
- [17] Athens Technology Centre, Engineering Ingegneria Informatica, University of Lancaster, e University of Athens. Ensuring Dependability of P2P Applications at Architectural Level, 2002. P2P Architect Project. Disponível em <http://www.atc.gr/p2p_architect/results/0101F05_P2P_Survey.pdf>. Acessado em Janeiro de 2005.
- [18] Qian Chen, Hyunseok Chang, Ramesh Govindan, Sugih Jamin, Scott Shenker, e Walter Willinger. The Origin of Power-Laws in Internet Topologies Revisited. *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'2002)*, 2002.
- [19] Imrich Chlamtac, Marco Conti, e Jennifer Liu. Mobile Ad Hoc Networking: Imperatives and Challenges. *Ad Hoc Networks Journal*, 1:13–64, 2003.
- [20] Romit Roy Choudhury, Krishna Paul, e Somprakash Bandyopadhyay. MARP: A Multi-Agent Routing Protocol for Mobile Wireless Ad Hoc Networks. *Autonomous Agents and Multi-Agent Systems*, 8(1):47–68, 2004.
- [21] Alberto Colorni, Marco Dorigo, e Vittorio Maniezzo. Distributed Optimization by Ant Colonies. *Proceedings of the First European Conference on Artificial Life*, pág. 134–142. Elsevier Publishing, 1992.
- [22] Microsoft Corporation. MSN Messenger Home Page. Disponível em <<http://messenger.msn.com>>. Acessado em Janeiro de 2005.
- [23] D'Agents: Mobile Agents at Dartmouth College. Disponível em <<http://agent.cs.dartmouth.edu>>. Acessado em Janeiro de 2005.
- [24] Prithviraj Dasgupta. Improving Peer-to-Peer Resource Discovery Using Mobile Agent Based Referrals. *Proceedings of the 2nd Workshop on Agent Enabled P2P Computing*, pág. 186–197, 2003.
- [25] Kalyanmoy Deb, editor. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2004)*, Seattle, USA, 2004. Springer.

- [26] Marco Dorigo, Vittorio Maniezzo, e Alberto Coloni. The Ant System: Optimization by a Colony of Cooperating Agents. *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*, 26(1):29–41, 1996.
- [27] Margery J. Doyle e Michael L. Kalish. Stigmergy: Indirect Communication in Multiple Mobile Autonomous Agents. *Proceedings of the International Conference on Knowledge Systems and Coalition Operations (KSCO'2004)*, 2004.
- [28] Christian Erfurth, Peter Braun, e Wilhelm R. Rossak. Some Thoughts on Migration Intelligence for Mobile Agents. Relatório Técnico 9, Friedrich-Schiller-Universität Jena, Institut für Informatik, Abril de 2001.
- [29] Jeffrey M. Bradshaw et. al. *Handbook of Intelligent Information Technology*, capítulo Making Agents Acceptable to People. IOS Press, Amsterdam, Netherlands, 2003.
- [30] Niranjana Suri et. al. NOMADS: Toward an Environment for Strong and Safe Agent Mobility. *Proceedings of the Fourth International Conference on Autonomous Agents*, págs. 163–164, Barcelona, Spain, 2000. ACM Press.
- [31] Michalis Faloutsos, Petros Faloutsos, e Christos Faloutsos. On Power-Law Relationships of the Internet Topology. *Proceedings of the ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM'99)*, págs. 251–262, Cambridge, Massachusetts, USA, 1999. ACM Press.
- [32] Paul J. Feltovich, Jeffrey M. Bradshaw, Renia Jeffers, Niranjana Suri, e Andrzej Uszok. Social Order and Adaptability in Animal and Human Cultures as Analogues for Agent Communities: Toward a Policy-Based Approach. *Engineering Societies in the Agents World IV, 4th International Workshop (ESAW'2003)*, págs. 21–48, London, UK, 2003.
- [33] Foundation for Intelligent Physical Agents. The foundation for intelligent physical agents. Disponível em <<http://www.fipa.org>>. Acessado em Janeiro de 2005.
- [34] Alliance for Telecommunications Industry Solutions. American National Standard for Telecommunications - Telecom Glossary 2000, Fevereiro de 2001. Disponível em <<http://www.atis.org/tg2k/>>. Acessado em Janeiro de 2005.
- [35] George H. Forman e John Zahorjan. The Challenges of Mobile Computing. *IEEE Computer*, 27(4):38–47, 1994.
- [36] Stan Franklin e Art Graesser. Is it an Agent, or Just a Program?: A Taxonomy for Autonomous Agents. *Proceedings of the Third Workshop on Agent Theories, Architectures, and Languages (ECAI'96)*, págs. 21–35. Springer-Verlag, 1997.
- [37] James A. Freebersyser e Barry Leiner. *Ad Hoc Networking*, capítulo 2: A DoD Perspective on Mobile Ad Hoc Networks, págs. 29–51. Addison-Wesley Longman Publishing, 2001.
- [38] The Freenet Project Home Page. Disponível em <<http://freenet.sourceforge.net>>. Acessado em Janeiro de 2005.
- [39] Alfonso Fuggetta, Gian Pietro Picco, e Giovanni Vigna. Understanding Code Mobility. *IEEE Transactions on Software Engineering*, 24(5):342–361, 1998.

- [40] Prasanna Ganesan, Qixiang Sun, e Hector Garcia-Molina. YAPPERS: A Peer-to-Peer Lookup Service over Arbitrary Topology. *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'2003)*, 2003.
- [41] D. Gavalas, D. Greenwood, M. Ghanbari, e M. O'Mahony. An Infrastructure for Distributed and Dynamic Network Management Based on Mobile Agent Technology. *Proceedings of the IEEE International Conference on Communications (ICC'99)*, 1999.
- [42] P. P. Grassé. La Reconstruction Du Nid et les Coordinations Inter-Individuelles Chez Bellicositermes Natalensis et Cubitermes sp. La Theorie de la Stigmergie: Essai D'Interpretation des Termites Constructeurs. *Insect Societies*, 6:41–83, 1959.
- [43] Robert S. Gray. Soldiers, Agents and Wireless Networks: A Report on a Military Application. *Proceedings of the Fifth International Conference and Exhibition on the Practical Application of Intelligent Agents and Multi-Agents*, Manchester, England, Abril de 2000.
- [44] Robert S. Gray. Mobile Agents: Overcoming Early Hype and a Bad Name. *Proceedings of the 2004 IEEE International Conference on Mobile Data Management (MDM'04)*, pág. 302–303, 2004.
- [45] Robert S. Gray, David Kotz, George Cybenko, e Daniela Rus. *Handbook of Agent Technology*, capítulo Mobile Agents: Motivations and State-of-the-Art Systems. AAAI / MIT Press, 2000.
- [46] Fritz Hohl. The mobile agent list. Disponível em <<http://reinsburgstrasse.dyndns.org/mal/mal.html>>. Acessado em Janeiro de 2005.
- [47] Owen Holland e Chris Melhuish. Stigmergy, Self-Organization, and Sorting in Collective Robotics. *Artificial Life*, 5(2):173–202, 1999.
- [48] Bert Holldobler e Edward O. Wilson. *The Ants*. Belknap Press, 1990.
- [49] Michael N. Huhns e Larry M. Stephens. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, capítulo 2: Multiagent Systems and Societies of Agents, pág. 79–120. The MIT Press, 1999.
- [50] Cisco Systems Inc. *Internetworking Technologies Handbook*, capítulo 7: Network Management Basics. Cisco Press, 4th edition, 2003.
- [51] ICQ Inc. ICQ Home Page. Disponível em <<http://www.icq.com>>. Acessado em Janeiro de 2005.
- [52] Motorola Inc. Motorola Mesh Networks Technology. Disponível em <<http://www.meshnetworks.com>>. Acessado em Janeiro de 2005.
- [53] Object Management Group Inc. CORBA Website. Disponível em <<http://www.corba.org>>. Acessado em Janeiro de 2005.

- [54] Object Management Group Inc. Object Management Group Home Page. Disponível em <<http://www.omg.org>>. Acessado em Janeiro de 2005.
- [55] SPANWorks Inc. SPANworks Home Page. Disponível em <<http://www.spanworks.com>>. Acessado em Janeiro de 2005.
- [56] Sun Microsystems Inc. Java Technology. Disponível em <<http://java.sun.com>>. Acessado em Janeiro de 2005.
- [57] Nicholas R. Jennings, Katia Sycara, e Michael Wooldridge. A Roadmap of Agent Research and Development. *Autonomous Agents and Multi-Agent Systems*, 1(1):7–38, 1998.
- [58] Elias Procópio Duarte Jr. e Andréa Weber. A Distributed Network Connectivity Algorithm. *Proceedings of the Sixth IEEE International Symposium on Autonomous Decentralized Systems (ISADS'2003)*, pág. 285–292, Pisa, Italy, 2003.
- [59] Günter Karjoth, N. Asokan, e Ceki Gülcü. Protecting the Computation Results of Free-Roaming Agents. *Proceedings of the Second International Workshop on Mobile Agents (MA'99)*, pág. 195–207. Springer-Verlag, 1999.
- [60] Irene Katzela e Mischa Schwartz. Schemes for Fault Identification in Communication Networks. *IEEE/ACM Transactions on Networking*, 3:753–764, 1995.
- [61] James Kennedy e Russell C. Eberhart. *Swarm Intelligence*. Morgan Kaufmann Publishers Inc., 2001.
- [62] David Kotz e Robert S. Gray. Mobile Agents and the Future of the Internet. *SIGOPS — ACM Operating Systems Review*, 33(3):7–13, 1999.
- [63] David Kotz, Robert S. Gray, e Daniela Rus. Future Directions for Mobile Agent Research. *IEEE Distributed Systems Online*, 3(8), 2002.
- [64] Magnus Ljungberg e Andrew Lucas. The OASIS air-traffic management system. *Proceedings of the Second Pacific Rim International Conference on Artificial Intelligence (PRICAI'92)*, Seoul, Korea, 1992.
- [65] Paulo Afonso Lopes. *Probabilidades & Estatística*. Reichmann & Affonso Editores, Rio de Janeiro, 1999.
- [66] Massive Ltd. Massive software home page. Disponível em <<http://www.massivesoftware.com>>. Acessado em Fev 2005.
- [67] Qin Lv, Pei Cao, Edith Cohen, Kai Li, e Scott Shenker. Search and Replication in Unstructured Peer-to-Peer Networks. *Proceedings of the 16th International Conference on Supercomputing (ICS'02)*, pág. 84–95, New York, USA, 2002. ACM Press.
- [68] Joseph P. Macker e M. Scott Corson. *Ad Hoc Networking*, capítulo 9: Mobile Ad Hoc Networks (MANETs): Routing Technology for Dynamic, Wireless Networks. Wiley Publishers, New York, USA, 2003.
- [69] G. Malkin. RIP Version 2, 2003. RFC 2453.

- [70] S. Marwaha, C. K. Tham, e D. Srinivasan. A Novel Routing Protocol Using Mobile Agents and Reactive Route Discovery for Ad-Hoc Wireless Networks. *Proceedings of IEEE International Conference on Networks 2002 (ICON'2002)*, 2002.
- [71] Alberto Medina, Ibrahim Matta, e John Byers. On the Origin of Power Laws in Internet Topologies. *SIGCOMM Computer Communication Review*, 30(2):18–28, 2000.
- [72] Sun Microsystems. JXTA.org. Disponível em <<http://www.jxta.org>>. Acessado em Janeiro de 2005.
- [73] Nelson Minar, Kwindla Hultman Kramer, e Pattie Maes. *Software Agents for Future Communication Systems*, capítulo 12: Cooperating Mobile Agents for Dynamic Network Routing, pág. 287–304. Springer-Verlag, 1999.
- [74] LLC. Napster. Napster Home Page. Disponível em <<http://www.napster.com>>. Acessado em Janeiro de 2005.
- [75] Sharman Networks. Sharman Networks Home Page. Disponível em <<http://www.sharmannetworks.com>>. Acessado em Janeiro de 2005.
- [76] Institute of Electrical e Electronics Engineers Inc. (IEEE). The Working Group for Wireless LAN Standards Home Page. Disponível em <<http://grouper.ieee.org/groups/802/11/>>. Acessado em Janeiro de 2005.
- [77] LLC OSMB. Gnutella.com. Disponível em <<http://www.gnutella.com>>. Acessado em Janeiro de 2005.
- [78] Norman H. Packard e Mark A. Bedau. *Encyclopedia of Cognitive Science*, capítulo Artificial Life. Macmillan Publishers, London, UK, 2000.
- [79] Manoj Parameswaran, Anjana Susarla, e Andrew B. Whinston. P2P Networking: An Information-Sharing Alternative. *IEEE Computer*, 34(7):31–38, 2001.
- [80] Chris A. Parker, H. Zang, e C. Ronald Kube. Blind Bulldozing: Multiple Robot Nest Construction. *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS'2003)*, pág. 2010–2015, 2003.
- [81] Larry L. Peterson e Bruce S. Davie. *Computer Networks: A Systems Approach*. Morgan Kaufmann Publishers Inc., 3rd edition, 2003.
- [82] Gian Pietro Picco. muCode Home Page. Disponível em <<http://mucode.sourceforge.net>>. Acessado em Janeiro de 2005.
- [83] Gian Pietro Picco. *Understanding, Evaluating, Formalizing, and Exploiting Code Mobility*. Tese de Doutorado, Politecnico di Torino, Italy, February de 1998.
- [84] Marius Portmann e Aruna Seneviratne. Cost-Effective Broadcast for Fully Decentralized Peer-to-Peer Networks. *Computer Communications*, 26(11):1159–1167, 2003.
- [85] A. Puliafito e O. Tomarchio. Using Mobile Agents to Implement Flexible Network Management Strategies. *Computer Communications Journal*, 23:708–719, 2000.

- [86] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, e Scott Schenker. A Scalable Content-Addressable Network. *Proceedings of the 2001 ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM'2001)*, pág. 161–172. ACM Press, 2001.
- [87] Craig W. Reynolds. Interaction with Groups of Autonomous Characters. *Proceedings of the 2000 Game Developers Conference*, EUA, 2000.
- [88] Matei Ripeanu, Ian Foster, e Adriana Iamnitchi. Mapping the Gnutella Network: Properties of Large-Scale Peer-to-Peer Systems and Implications for System Design. *IEEE Internet Computing Journal*, 6(1):50–57, 2002.
- [89] Ichiro Satoh. A Framework for Building Reusable Mobile Agents for Network Management. *Proceedings of IEEE/IFIP International Network Operations and Management Symposium (NOMS2002)*, 2002.
- [90] Ruud Schoonderwoerd, Owen Holland, e Janet Bruten. Ant-Like Agents for Load Balancing in Telecommunications Networks. *Proceedings of the First ACM International Conference on Autonomous Agents (AGENTS'97)*, pág. 209–216, California, USA, 1997. ACM Press.
- [91] Search Engine Watch Home Page. Disponível <<http://www.searchenginewatch.com>>. Acessado em Janeiro de 2005.
- [92] Ion Stoica, Robert Morris, David R. Karger, M. Frans Kaashoek, e Hari Balakrishnan. Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications. *Proceedings of the 2001 ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM'2001)*, pág. 149–160, 2001.
- [93] Gatot Susilo, Andrzej Bieszczad, e Bernard Pagurek. Infrastructure for Advanced Network Management based on Mobile Code. *Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS'98)*, New Orleans, USA, Fevereiro de 1998.
- [94] Andrew S. Tanenbaum. *Computer Networks*. Prendice-Hall, 4th edition, 2002.
- [95] Hongsuda Tangmunarunkit, Ramesh Govindan, Sugih Jamin, Scott Shenker, e Walter Willinger. Network Topology Generators: Degree-Based vs. Structural. *Proceedings of the ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM'2002)*, pág. 147–159, 2002.
- [96] Mark Ward. There's an Ant in My Phone! *New Scientist*, pág. 32–35, Janeiro de 1998. Disponível em <<http://www.newscientist.com/article.ns?id=mg15721184.900>>. Acessado em Fevereiro de 2005.
- [97] Gerhard Weiss, editor. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. The MIT Press, 1999.
- [98] Tony White. Swarm Intelligence and Problem Solving in Telecommunications. *Canadian Artificial Intelligence Magazine*, pág. 14–16, Primavera de 1997.

- [99] Tony White, Andrzej Bieszczad, e Bernard Pagurek. Distributed Fault Location in Networks Using Mobile Agents. *Proceedings of the 3rd International Workshop on Agents in Telecommunication Applications (IATA '98)*, France, 1998.
- [100] Tony White, Andrzej Bieszczad, Bernard Pagurek, G. Sugar, e X. Tran. Intelligent Network Modeling using Mobile Agents. *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM'98)*, pág. 1082–1087, 1998.
- [101] Tony White e Bernard Pagurek. Towards Multi-Swarm Problem Solving in Networks. *Proceedings of the 3rd International Conference on Multi Agent Systems (ICMAS'98)*. IEEE Computer Society, 1998.
- [102] Uwe G. Wilhelm, Sebastian Staamann, e Levente Buttyán. Introducing Trusted Third Parties to the Mobile Agent Paradigm. *Secure Internet Programming: Security Issues for Mobile and Distributed Objects, Lecture Notes in Computer Science*, 1603:469–489, 1999.
- [103] Edward O. Wilson. *Sociobiology: The New Synthesis*. Harvard University Press, 1975.
- [104] Michael Wooldridge. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, capítulo 1: Intelligent Agents, pág. 27–77. The MIT Press, 1999.
- [105] B. Yang e H. Garcia-Molina. Efficient Search in Peer-to-Peer Networks. *Proceedings of the 22nd IEEE International Conference on Distributed Computing Systems (ICDCS'2002)*, 2002.
- [106] Özalp Babaoglu, Hein Meling, e Alberto Montresor. Anthill: A Framework for the Development of Agent-Based Peer-to-Peer Systems. *Proceedings of the 22nd International Conference on Distributed Computing Systems (ICDCS'02)*, Vienna, Austria, 2002. IEEE Computer Society.
- [107] B. Y. Zhao, J. D. Kubiatowicz, e A. D. Joseph. Tapestry: An Infrastructure for Fault-tolerant Wide-area Location and Routing. Relatório Técnico UCB/CSD-01-1141, Computer Science Department, University of California, Berkeley, 2001.